IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
REQUEST FOR FILING APPLICATION UNDER 37 CFR 1.53(b)
WITHOUT FILING FEE OR EXECUTED INVENTOR'S DECLARATION

Assistant Commissioner for Patents
Washington, DC 20231

Atty. Dkt. 723-939

Date: October 3, 2000

Sir:

This is a request for filing a new PATENT APPLICATION under Rule 53(b) entitled:

GAME SYSTEM AND GAME INFORMATION STORAGE MEDIUM USED FOR SAME

without a filing fee and/or without an executed inventor's oath/declaration.

This application is made by the below identified inventor(s). Attached hereto are the following papers:

☒ An abstract together with

71 pages of specification and claims including

21 numbered claims and also attached are

54 sheets of accompanying drawings.

☒ This application is based on the following prior foreign application(s):

Application No.	Country	Filing Date
11-282592	JAPAN	4 October 1999
2000-174573	JAPAN	9 June 2000

respectively, the entire content of which is hereby incorporated by reference in this application, and priority is hereby claimed therefrom.

☐ This application is based on the following prior provisional application(s):

Application No.   **Filing Date**

respectively, the entire content of which is hereby incorporated by reference in this application, and priority is hereby claimed therefrom.

☐ Certified copy/ies of foreign applications attached.

☐ This application is a ☐ continuation/☐ division/☐ continuation-in-part of application Serial No.   , filed   .

☐ Please amend the specification by inserting before the first line: --This application is a ☐ continuation/☐ division/ ☐ continuation-in-part of application Serial No.   , filed   , the entire content of which is hereby incorporated by reference in this application.--

☐ Please amend the specification by inserting before the first line: --This is a continuation of PCT application No.   , filed   , the entire content of which is hereby incorporated by reference in this application.--

☐ Please amend the specification by inserting before the first line: --This application claims the benefit of U.S. Provisional Application No.   , filed   , the entire content of which is hereby incorporated by reference in this application.--

☒ Preliminary amendment to claims (attached hereto), to be entered before calculation of the fee.

☐ Also attached.

1. Inventor: Iwao (first)   MI   MASUYAMA (last)   Japan (citizenship)
   Residence: (city) Kyoto   (state/country) Japan
   Post Office Address: c/o Nintendo Co., Ltd. 60, Fukuine Kamitakamatsu-cho, Higashiyama-ku, Kyoto, Japan
   (Zip Code)

2. Inventor: Toshiaki (first)   MI   SUZUKI (last)   Japan (citizenship)
   Residence: (city) Kyoto   (state/country) Japan
   Post Office Address: c/o Nintendo Co., Ltd. 60, Fukuine Kamitakamatsu-cho, Higashiyama-ku, Kyoto, Japan
   (Zip Code)

NOTE: FOR ADDITIONAL INVENTORS, check box ☒ and attach sheet with same information.

Address all future communications to NIXON & VANDERHYE P.C., 1100 North Glebe Road, 8th Floor, Arlington, Virginia 22201.

1100 North Glebe Road, 8th Floor
Arlington, Virginia 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100
MEN:ms

NIXON & VANDERHYE P.C.
By Atty: Mark E. Nusbaum, Reg. No. 32,348

Signature:

463510

3.　　Inventor:　　　Akira　　　　　　　　　　　　　TAHARA　　　　　　Japan
　　　　　　　　　　　　(first)　　　　MI　　　　　　　(last)　　　　　(citizenship)
　　　Residence: (city)　Kyoto　　　　　　　　　　(state/country) Japan
　　　Post Office Address: c/o Nintendo Co., Ltd. 60, Fukuine Kamitakamatsu-cho, Higashiyama-ku, Kyoto, Japan
　　　　　　(Zip Code)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

MASUYAMA et al.   Atty. Ref.: 723-939

Serial No. (To Be Assigned)   Group:

Filed: October 3, 2000   Examiner:

For: GAME SYSTEM AND GAME INFORMATION
     STORAGE MEDIUM USED FOR SAME

* * * * * * * * * * *

October 3, 2000

Assistant Commissioner for Patents
Washington, DC 20231
Sir:

PRELIMINARY AMENDMENT

In order to place the above-identified application in better condition for examination, please amend the application as follows:

IN THE CLAIMS

Claim 5, line 1, change "any of claims 1 to 4" to --claim 1--.
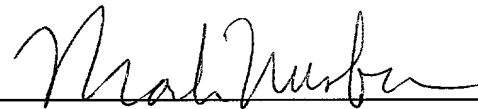
REMARKS

The above amendments are made to place the claims in a more traditional format.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____
                 Mark E. Nusbaum
                 Reg. No. 32,348

MEN:ms
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

U.S. PATENT APPLICATION

Inventor(s):   Iwao MASUYAMA
Toshiaki SUZUKI
Akira TAHARA

Invention: GAME SYSTEM AND GAME INFORMATION STORAGE MEDIUM USED
FOR SAME

SPECIFICATION

TITLE OF THE INVENTION

Game System and Game Information Storage Medium Used for Same

BACKGROUND OF THE INVENTION

5   Field of the invention

This invention relates to a game system and game information storage medium used for same. More particularly, the invention relates to a game system and game information storage medium used for same, which detects a change amount and direction of a tilt, movement or impact applied to a housing of a portable game apparatus or to a

10   controller of a video game apparatus.

Description of the prior art

In operating the conventional portable game apparatus, a player manipulates the operation switch, such as a direction instructing key (joystick) or buttons while holding the video game machine's controller (controller housing) or portable game apparatus'

15   housing by both hands. For example, if the player presses a direction instructing key at any one of up, down, left and right pressing points, a moving (player-controlled) character is moved in the pressed direction of up, down, left or right. If the action button is operated, the moving character is changed in state of display, e.g. the moving character is caused in action, such as jump, as defined on the action button.

20   Also, in the conventional game apparatus or game software (game information storage medium), the player can operate the operation switch in order to change the motion of a moving (player-controlled) character acting as a player's other self on the screen. Consequently, it has been difficult for the player to change the game space (or background scene) freely through his or her manipulation.

25   In the conventional game operation method, the player has been required to

remember the way to operate a game according to the suggestion given in the game-software instruction manuals.   Furthermore, the use of general-purpose operation switch has made it difficult to realize a change of the game space (or game scene) in a manner matched to an operation-switch manipulation feeling of the player, resulting in mismatch between operation feeling and screen-display state.   Under such situations, the player possibly has encountered difficulty in concentrating on a game play before mastering the way to manipulate the game, losing his or her interest.

Meanwhile, with the conventional game apparatus or game software, the game space (or background scene) could not have been changed by player's operation, thus insufficient in game space variation and hence amusement.

SUMMARY OF THE INVENTION

Therefore, it is a primary object of the invention to provide a game system and game information storage medium used for same which can change the state of a game space according to operation by a player.

Another object of the invention is to provide a game system and game information storage medium used for same which can change the state of a game space through simple operation so that a player can concentrate on game play with enhanced enthusiasm without the necessity of skill on operation way.

Still another object of the invention is to provide a game system and game information storage medium used for same which can realize the change of a game scene matched to an operation feeling through the match between player's operation and game-space change.

Yet another object of the invention is to provide a game system and game information storage medium used for same which can change the state of a game space

- 2 -

through the interaction with a plurality of portable game apparatuses to allow a plurality of players to cooperate or compete with thereby providing a variety of game-space change states, enhanced interest of game and virtual reality amusement.

A first invention (invention of claim 1) is a game system having in a related fashion, to a game apparatus having game program storage means storing a game program and processing means for executing the game program, display means to display an image based on a result of processing by the processing means. Provided are a housing to be held by a player and change-state detecting means. The change-state detecting means is provided related to the housing and detects at least one of an amount (e.g. tilt amount, movement amount, impact amount or the like) and a direction (e.g. tilt direction, movement direction, impact direction or the like) of a change applied to the housing. The game program storage means stores game space data, a display control program and a simulation program.

The game space data includes image data to display a space for game play. A display control program causes the display means to display a game space based on the game space data. A simulation program simulates based on an output of the change-state detecting means such that a state of the game space is changed related to at least one of a change amount and a change direction applied to the housing.

Here, game space means a world of a game that the game is possible to play, and is different by game kind or genre and presented to a player through a display screen. For example, for an action or roll-playing game having a moving (player-controlled) character to move therein, game space may be a background, a maze or other maps. For a battle game, it may be a ring (in addition to this, included is a space of audience seats and the above of ring). For a race game, it may be a space of a course for running and a periphery of the course. For a shooting game, a background scene such as a cosmic space

- 3 -

for a background of a character (however, characters are not requisite, and a game space no character exists is to be contemplated). In a game using a tool, game space may be a scene to associate use of a tool.

Simulation refers to game control for analogously representing a change caused in the actual space in a form of a game-space state change, based on at least one of an amount and a direction of a tilt, movement or impact applied to the housing. Game control includes the case of simulation on a state change of the game space itself and the case of simulation of an indirect effect upon another object caused due to a change in state of the game space. The former is a case that simulation is made such that, when an impact is given to the housing, a land in the game space is transformed on an assumption that energy has been supplied to the game space. The latter is a case that simulation is made such that, when the housing is tilted, a ball existing on a maze plate rolls on an assumption that the maze plate as an example of a game space is tilted. Where simulating a state change of a game space, it is possible to consider a case of varying a parameter such as of temperature rise in the game space, in addition to the case of causing a change of display including land transformation.

A second invention (invention of claim 15) is a game information storage medium storing a game program and detachably loaded in a game system structured by operating means having display means in a related manner and including a housing to be held by a player, change-state detecting means provided related to the housing and for detecting at least one of an amount and a direction of a change applied to the housing, and processing means to display on the display means an image obtained by processing a program. The game information storage medium stores game space data, a display control program and a simulation program.

The game space data includes image data to display a space for game play. A

display control program causes the display means a game space based on the game space data. A simulation program provides simulation based on an output of the change-state detecting means such that a state of the game space is changed related to at least one of an amount and a direction of a change applied to the housing.

A third invention (invention of claim 16) is a game information storage medium storing a game program and detachably loaded in a portable game apparatus including a housing integrally having display means to be held by a player, and processing means to display on the display means an image obtained by processing a program, wherein a change-state detecting means is provided related to one of the portable game apparatus and the game information storage medium and for detecting at least one of an amount and a direction of a change applied to one of a housing of the portable game apparatus and the game information storage medium.

The game information storage medium stores game space data, a display control program and a simulation program. The game space data includes image data to display a space for game play. A display control program causes the display means to display a game space based on the game space data. A simulation program provides simulation based on an output of the change-state detecting means such that a state of the game space is changed related to at least one of an amount and a direction of a change applied to the housing.

A fourth invention (invention of claim 18) is a game system structured at least by two game apparatuses to be interacted with each other. The two game apparatuses each have game program storage means to store a program, processing means to execute a game program, and a housing to be held by a player, and in a related fashion display means to display an image based on a result of processing by the processing means. At least one of the two game apparatuses is provided related to the housing and having

change-state detecting means to detect at least one of an amount and a direction of a change applied to the housing. The game system further having data transmitting means connected to the two game apparatuses and for transmitting mutually-related data to the game apparatus on the opposite side.

5 The respective of the game program storage means of the two game apparatuses store game space data and display control programs. The game space data includes image data to display a space for game play. The display control program to cause the display means to display a game space based on the game space data. The game program storage means of at least the other of the two game apparatuses further includes a simulation

10 program to provide simulation based on an output of the change-state detecting means of the one game apparatus transmitted through the data transmitting means such that a state of the game space of the other of the game apparatuses is changed related to at least one of an amount and a direction of a change applied to the housing of one of the game apparatuses.

15 According to this invention, it is possible to obtain a game system and game information storage medium used for same that can change a state of a game space.

Also, according to the invention, a game system and game information storage medium used for same is to be obtained which can change the state of a game space through simple operation so that a player can concentrate on game play with enhanced

20 enthusiasm without the necessity of skill on operation ways.

Also, according to the invention, a game system and game information storage medium used for same is to be obtained which can realize the change of a game scene matched to an operation feeling through the match between player's operation and game-space change.

25 Further, according to the invention, a game system and game information storage

medium used for same is to be obtained which can change the state of a game space through the interaction with a plurality of portable game apparatuses to allow a plurality of players to cooperate or compete with thereby providing a variety of game-space change states, enhanced interest of game and virtual reality amusement.

5   The above described objects and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

10   Figure 1 is an external view of a portable game apparatus of one embodiment of the present invention;

Figure 2 is a view showing a definition of XYZ axes;

Figure 3 is a block diagram of the portable game apparatus;

Figure 4 is a block diagram of a sensor interface;

15   Figure 5 is a diagram showing a principle on measuring the output of an acceleration sensor;

Figure 6 is a view showing a structure of a Z-axis contact switch;

Figure 7 is a view showing that a movement input (or impact input) in the Z-axis direction is detected by the Z-axis contact switch;

20   Figure 8 is an example of a game scene of a first embodiment;

Figure 9 is an illustrative view showing a slide input;

Figure 10 is an illustrative view showing a tilt input;

Figure 11 is an illustrative view showing an impact input in an X-axis or Y-axis direction;

25   Figure 12 is an illustrative view showing a movement input (impact input) in the

Z-axis direction;

Figure 13 is an illustrative view showing a way to utilize a slide input;

Figure 14 is an illustrative view showing a way to utilize a tilt input;

Figure 15 is an illustrative view showing a way to utilize an impact input;

Figure 16 is a memory map of a program ROM of the first embodiment;

Figure 17 is a memory map of a work RAM of the first embodiment;

Figure 18 is a memory map of a display RAM of the first embodiment;

Figure 19 is a memory map of a backup RAM of the first embodiment;

Figure 20 is an acceleration-sensor output conversion table of the first embodiment;

Figure 21 is an acceleration-sensor output conversion table of the first embodiment;

Figure 22 is an acceleration-sensor output conversion table of the first embodiment;

Figure 23 is an acceleration-sensor output conversion table of the first embodiment;

Figure 24 is an acceleration-sensor output conversion table of the first embodiment;

Figure 25 is an acceleration-sensor output conversion table of the first embodiment;

Figure 26 is an acceleration-sensor output conversion table of the first embodiment;

Figure 27 is a main routine flowchart of the first embodiment;

Figure 28 is a 0G set process flowchart of the first embodiment;

Figure 29 is a neutral-position set process flowchart of the first embodiment;

Figure 30 is a game map elect process flowchart of the first embodiment;

Figure 31 is a sensor output read process flowchart of the first embodiment;

Figure 32 is an each-object moving process flowchart of the first embodiment;

Figure 33 is a player-character moving process flowchart of the first embodiment;

Figure 34 is an NPC moving process flowchart of the first embodiment;

Figure 35 is a jump moving process flowchart of the first embodiment;

Figure 36 is a wave moving process flowchart of the first embodiment;

Figure 37 is a collision process flowchart of the first embodiment;

Figure 38 is a screen-scroll explanatory view (before scroll) of the first embodiment;

Figure 39 is a screen-scroll explanatory view (after scroll) of the first embodiment;

Figure 40 is a screen-scroll process flowchart of the first embodiment;

Figure 41 is an example of a game scene of a second embodiment;

Figure 42 is an example of a game scene (land-upheaval process) of the second embodiment;

Figure 43 is an example of a game scene (range-of-sight moving process) of the second embodiment;

Figure 44 is an example of a game scene (temperature increasing process) of the second embodiment;

Figure 45 is a memory map of a program ROM of the second embodiment;

Figure 46 is a memory map of a work RAM of the second embodiment;

Figure 47 is a main routine flowchart of the second embodiment;

Figure 48 is a range-of-sight moving process flowchart of the second embodiment;

Figure 49 is a land-upheaval process flowchart of the second embodiment;

Figure 50 is an example of a game scene of a third embodiment;

Figure 51 is an example of a game scene (frypan space process) of the third embodiment;

Figure 52 is an example of a game scene (frypan space process) of the third embodiment;

5   Figure 53 is an example of a game scene (kitchen-knife space process) of the third embodiment;

Figure 54 is a memory map of a work RAM of the third embodiment;

Figure 55 is a main routine flowchart of the third embodiment;

Figure 56 is a frypan space process flowchart of the third embodiment;

10   Figure 57 is a kitchen-knife space process flowchart of the third embodiment;

Figure 58 is an egg jump process flowchart of the third embodiment;

Figure 59 is cabbage cut process flowchart of the third embodiment;

Figure 60 is an example of a game scene of a fourth embodiment;

Figure 61 is a main routine flowchart of a game apparatus 10 of a fourth

15   embodiment;

Figure 62 is a main routine flowchart of a game apparatus 40 of the fourth embodiment;

Figure 63 is a mater-unit map confirming process flowchart of the fourth embodiment;

20   Figure 64 is a slave-machine map confirming process flowchart of the fourth embodiment;

Figure 65 is a master-machine communication interrupt process flowchart of the Figure 4 embodiment;

Figure 66 is a slave-machine communication interrupt process flowchart of the

25   Figure 4 embodiment;

Figure 67 is an example that the invention is applied to a controller of a home-use game apparatus; and

Figure 68 is an example of a scene that the invention is applied to a controller of a home-use game apparatus.

5

BRIEF DESCRIPTION OF THE PREFERRED EMBODIMENTS

(First Embodiment)

With reference to Figure 1 to Figure 40, explanations will be made on a portable game apparatus according to a first embodiment of the present invention. Figure 1 is an

10 outside view showing a portable game apparatus. The portable game apparatus includes a game machine main body 10 and a game cartridge (hereinafter referred merely to as "cartridge") 30 to be unloadably loaded on the game machine main body 10. The cartridge 30, when loaded on the game machine main body 10, is put in electrical connection to the game machine main body. The game machine main body 10 is

15 provided with a housing 11. The housing 11 includes therein a board having circuits configured as shown in Figure 3, hereinafter described. The housing 11 has, on one main surface, a LCD 12 and operation keys 13a – 13e and, on the other surface, a hole (cartridge insertion hole) 14 formed to receive a cartridge 30. A connector 15 is provided on a side surface, to allow connection with a communication cable for communication, as

20 required, with other portable game apparatuses.

Figure 2 is an illustrative view showing a relationship between the portable game apparatus and XYZ axes thereon. In a state the portable game apparatus is positioned with the LCD 12 directed up and the operation switches positioned toward this, an X-axis is taken in a horizontal direction of the portable game apparatus (a plus direction taken rightward), an Y-axis is in a vertical direction (a plus direction taken depthwise), and a

25

Z-axis is in a thickness direction (a plus direction taken upward).

Figure 3 is a block diagram of the portable game apparatus. The game machine main body 10 incorporates a board 27 therein. The board 27 is mounted with a CPU 21. The CPU 21 is connected with a LCD driver 22, an operation key 13, a sound generator circuit 23, a communication interface 24, a display RAM 156 and a work RAM 26. The sound generator circuit 23 is connected with a speaker 16. The communication interface 24 is to be connected to another portable game apparatus 40 through a connector 15 and communication cable 50. Note that, although the communication method with the other portable game apparatus 40 was shown by a method using the communication cable 50, it may use radio communication, handy phone or the like.

The cartridge 30 incorporates a board 36. The board 36 is mounted with a program ROM 34 storing a game program and game data, hereinafter described with reference to Figure 16, and a backup RAM 35 storing a game data, hereinafter described with reference to Figure 19. In addition to these storage means, the cartridge 30 includes, as one example of detecting means for detecting tilt, movement and impact to the portable game apparatus main body, an XY-axis acceleration sensor 31 to detect accelerations in X-axis and Y-axis directions and a Z-axis contact switch 32 to detect an acceleration in a Z-axis direction. Also, the cartridge 30 includes a sensor interface 33 as an interface to the acceleration detecting means. Where using a triaxial acceleration sensor capable of detecting accelerations in all the X-axis, Y-axis and Z-axis directions, the Z-axis contact switch 32 will be unnecessary. Incidentally, the biaxial acceleration sensor (XY-axis acceleration sensor) is more inexpensive than that sensor. Because this embodiment does not require high accuracy of acceleration detection in the Z-axis direction, a Z-axis contact switch 32 is employed that is simple in structure and cheap in price. Also, where high accuracy is not required in the XY-axis direction, detecting means having a similar

structure to the Z-axis contact switch may be used in detecting an acceleration in the XY-axis direction.

The program ROM 34 is stored with a game program to be executed by a CPU 21. The work RAM 26 is stored with temporary data required to execute the game program. The backup RAM 35 is to store game data to be kept memorized even where a power to the portable game apparatus be turned off. The display data obtained through executing the game program by the CPU 21 is stored in the display RAM 25, which can be displayed on the LCD 12 through a LCD driver 22. Similarly, the sound data obtained through executing the game program by the CPU 21 is delivered to the sound generator circuit 23 so that sound is generated as effected sound through the speaker 16. Operation switches 13 are for game operation. However, the operation key 13 is auxiliary one as far as the present embodiment is concerned. The player is allowed to operate for game play principally by tilting or moving or giving impact to the portable game apparatus. The tilt, movement and impact to the portable game apparatus during game operation are to be detected by the XY-axis acceleration sensor 31 and Z-axis contact switch 12. The CPU 21 can execute the game program by utilizing the output values of the acceleration detecting means,.

For a game with using a plurality of portable game apparatuses, the data obtained through executing a game program by the CPU 21 is delivered to the communication interface 24 and then sent to another portable game apparatus 40 via the connector 15 and communication cable 50. Meanwhile, the game data of the other portable game apparatus 40 comes to the CPU 21 through the communication cable 50, connector 15 and communication interface 24.

Figure 4 is a detailed block diagram of the sensor interface 33. The sensor interface 33 includes an X counter 331, a Y counter 332, a count stop circuit 33, latches

- 13 -

334 and 335, a decoder 336 and a general-purpose I/O port 337. The X counter 331

counts pulses of a clock signal Φ based on an XY-axis output of the acceleration sensor

31. The Y counter 332 counts pulses of the clock signal Φ based on a Y-axis output. The

count stop circuit 333 sends a count stop signal to the X counter 331 in response to a fall

5   in an X-axis output of the XY-axis acceleration sensor 31, and a count stop signal to the Y

counter 332 in response to a fall in the Y-axis output. The latches 334 and 335 hold

respective values of the X counter 331 and the Y counter 332. The decoder 336 transmits

a start/reset signal to the X counter 331, Y counter 332, latches 334 and 335. The

general-purpose I/O port 337 is used to connect with an extension unit. The latches 334

10   and 335 also hold an output value of the Z-axis contact switch 32 ("0" or "1").

Specifically, a highest order bit of the latch 334, 335 is assigned to an output value of the

Z-axis contact switch 32, while the remaining lower order bits are assigned to the values

of the X counter and Y counter. The extension units connectable to the general-purpose

I/O port 337 include a vibration unit which vibrates in relation to a game program

15   providing a game with a realism feeling.

Figure 5 is an illustrative view showing a principle that the sensor interface 33

measures a count value having a corresponding magnitude to an acceleration from an

output of the acceleration sensor 31. The acceleration sensor 31 in this embodiment

outputs a signal representative of an acceleration magnitude with a duty ratio changed

20   with respect to one period of a waveform (period 1). It is shown in this case that the

greater the ratio of a high level period (period 2 or period 3) within one period the greater

an acceleration has been detected. Also, the acceleration sensor 31 outputs a magnitude

of X-axis acceleration through its X-axis output and a magnitude of Y-axis acceleration

through the Y-axis output.

25   When a count start signal outputted from the decoder 336 becomes a low level, the

- 14 -

X counter 331 detects a rise from low to high level in the X-axis output and then starts counting. Specifically, the X counter 331 inches up its count value each time a clock signal Φ is given, and stops the counting in response to a count stop signal sent from the count stop circuit 333. In this manner, the X counter331 counts on the clock signal Φ during a period (period 2) of between a rise of an X-axis output to a high level and a fall of same to a low level, immediately after the count start signal has become a low level. The Y counter 332, similarly, counts on the clock signal Φ during a period (period 3) of between a rise of the Y-axis output to a high level and a fall of same to a low level, immediately after the count start signal has become low level. In this manner, the X counter 331 holds a count value dependent upon a magnitude of an X-axial acceleration while the Y counter 332 holds a count value dependent upon a magnitude of a Y-axial acceleration. The values of the X counter 331 and Y counter 332 are held in the latch 334 and latch 335 so that the data of latches 334 and 335 can be read out by the CPU 21 through the data bus and utilized for a game program.

The X counter 331 and the Y counter 332 each perform counting, for example, from "0" up to "31", wherein setting is made such that, with respect to a count value "15" as a reference (acceleration 0), - 2G (twice a gravity acceleration in a minus direction) is assigned by a count value "0" and 2G (twice the gravity acceleration in a plus direction) is by a count value "31". The CPU 21 reads in such a count value based on a game program wherein the count value "15" is read as "0", the count value "0" as "-15" and the count value "31" as "16". Accordingly, when the acceleration sensor 31 detects an acceleration in the minus direction, the CPU has a minus (-) reading value. When an acceleration in the plus direction is detected, the CPU has a plus (+) reading value.

Figure 6 shows a structural of the contact switch 32. The contact switch 32 is structured by a spherical contact 321, contacts 322 and 323, and a box member 324 which

are formed of a conductor. Specifically, the spherical contact 321 is movably held almost at a center of a space defined within the member 324. For this reason, the box member 324 has, in its inner bottom, a depression 324a at which the spherical contact 321 can be rested at almost the center. The box member 324 has, at above, sheet-formed contacts 322 and 323 having respective one ends formed with semicircular cut-outs 322a and 323a. The sheet contacts 322 and 323, at their other ends, are secured to a board 36 with the one ends opposed to each other. The box member 324 is fixedly held by the board 36 in a state of hung through the contact 322 and 323. With this structure, if the cartridge 30 is powerfully moved in the Z-axis direction (in a plus or minus direction), the spherical contact 321 shown in Figure 7 is moved in the Z-axis direction within the box member 324 and contacts with the contacts 322 and 323 simultaneously. Thus, the contact 322 and the contact 323 are conducted through the spherical contact 321, thereby detecting an acceleration input in the Z-axis direction. Based on a time for which the contact 322 and the contact 323 are in conduction, it is possible to detect a magnitude of acceleration in the Z-axis direction. Note that, when the cartridge 30 is moderately tilted, the spherical contact 321 moves in the box member 324 but does not short-circuit between the contacts 322 and 323, detecting no acceleration.

Figure 8 shows an example of a game scene. In this game scene, there are displayed a ball 61 as one example of a player character, tortoises 62 as one example of an enemy character (non-player character; hereinafter abbreviated as "NPC"), and a wall 63 and hole 64 forming a maze. Because a game map is a virtual map that is broader than a display range on an LCD 12, LCD 12 can display only part of the game map so that scroll is made in accordance with the movement of the player character 61. Although three tortoises 62a – 62c are displayed as NPC on the LCD 12, there exist many of other tortoises in the game map. Also, there exist on the game map such lands as floors, ice

surfaces, and under waters.

The ball 61 is changed in its moving amount or direction by player's operation, such as tilting of or applying movement or impact to the portable game apparatus. The shape is changed as required. Although the tortoise 62 is controlled of movement

5   (moved by self-control) by the game program, it is moved or changed in shape where the player tilts, moves or gives impact to the portable game apparatus.

Outlining this game, a player can manipulate the ball 61 on the game map with a maze formed by the walls 63, and smashes the tortoises 62a - 62c as an example of NPC. A tortoise if smashed will be vanished or erased away. If all the tortoises are successfully

10   vanished out of the game map, game clear is reached. There exist some holes 64 on the game map. If the ball 61 is fallen into the hole 64, one mistake is counted or the game becomes over.

Figure 9 to Figure 12 illustrate examples of game operation. Figure 9 illustrates a slide input in the X-axis or Y-axis direction. A movement (slide) in the X-axis direction

15   is detected based upon an X-axis output of the XY-axis acceleration sensor 31, and a movement (slide) in the Y-axis direction is detected based on a Y-axis output of the XY-axis acceleration sensor 31 (acceleration is caused by movement in the X-axis or Y-axis direction). Figure 10 illustrates a tilt input about the X or Y axis. A tilt about the X-axis is detected based on a Y-axis output of the XY-axis acceleration sensor 31, and a

20   tilt about the Y-axis is detected based upon an X-axis output of the XY-axis acceleration sensor 31 (a tilt if caused about the X-axis gives rise to acceleration in the Y-axis direction due to gravity, and a tilt if caused about the Y-axis causes acceleration in the X-axis direction due to gravity). Figure 11 illustrates an impact input in the X-axis or Y-axis direction. The acceleration input in the X-axis direction is outputted at an X-axis

25   output of the XY-axis acceleration sensor 31. In the case this output value is a constant

value or greater, it is considered that there has been an impact input. Also, the acceleration input in the Y-axis direction is outputted at a Y-axis output of the XY-axis acceleration sensor 31. In the case this output value is a constant value or greater, it is considered that there has been an impact input. Figure 12 illustrates a movement input (or impact input) in the Z-axis direction. The movement (or impact) in the Z-axis direction is detected by the Z-axis contact switch 32.

Figure 13 to Figure 15 illustrate an examples of a way to utilize the respective ones of game operation stated above. Figure 13 illustrates a way to utilize a slide input (as one example of a game scene in a game map select process hereinafter described with reference to Figure 30). In a case of displaying on the LCD 12 a part area of a virtual map broader than a display range of the LCD 12, the display area is scrolled by giving a slide input. Specifically, where providing a slide input in an X-axis plus direction, to be displayed is an area moved in the X-axis plus direction from the present display area. A slide input in the Y-axis direction is similarly processed to this. By thus processing a slide input, it is possible to provide a player with a feeling as if he or she peeps part of a vast world through the LCD 12. Incidentally, in this embodiment, such slide input alike this is merely utilized in a game map select process hereinafter described with reference to Figure 30, but not utilized in a game-map scroll process as a main game process. The way of processing to scroll a game map will be hereinafter described with reference to Figure 38 to Figure 40.

Figure 14 illustrates a way to utilize a tilt input about an X or Y axis. Where there is a tilt input about the X-axis, display is made such that a game character in a game scene (player character 61 and NPC 62) is moving parallel in the Y-axis direction (where tilting in a plus direction about the X-axis, display is made such that the game character is moving parallel in a Y-axis minus direction). Also, where there is a tilt input about the

Y-axis, display is made such that the game character in the game scene, player character 61 and NPC 62) is moving parallel in the X-axis direction (where tilting in a minus direction about the Y-axis, display is made such that the game character moves parallel in an X-axis minus direction). By thus processing a tilt input, it is possible to provide a player with a feeling as if a maze plate, as a game space, was tilted likewise the portable game apparatus and the game character was sliding (rolling) over the tilted maze plate. Incidentally, the game map includes lands, such as floor surface, ice surface and under-water, providing factors to vary a moving amount of the ball 61, so that an amount of movement is varied by a tilt input in a manner dependent upon a place where the game character is present. For example, the ball 61 is changed in magnitude of control in such a way that the movement amount is great on an ice surface easy to slide whereas the movement amount is small at under-water.

Figure 15 shows a way to utilize impact input or Z-axis movement input. When an impact input is applied in the X-axis or Y-axis direction, a different process is performed from the tilt input process (game character movement due to tilting the maze plate). For example, waves are caused in a water surface as a game space. When an impact input is applied in the X-axis plus direction, waves are caused in the X-axis plus direction. When an impact input is applied in an X-axis minus direction, waves are caused in the X-axis minus direction. This is true for an impact input in a Y-axis direction. Meanwhile, waves may be caused in a direction of a resultant vector of vector components, wherein an acceleration input in the X-axis direction is taken a vector component in the X-axis direction while an acceleration input in the Y-axis direction is a vector component in the Y-axis direction. The character is displayed as if it was carried away by the waves. The character may be put out of control while it is being carried by the waves. Also, when there is an input of movement in the Z-axis direction (or impact input), the ball 61 as one

- 19 -

example of a player character is displayed varying to make a jump. By thus processing the movement input in the Z-axis direction, the maze plate as a game space moves in the Z-axis direction in a way similar to the portable game machine. This can provide the player with a feeling as if the game character on the maze plate was caused to jump.

5   During the jump, the ball 61 will not move even if there is a tilt input. Also, when there is a movement input (or impact input) in the Z-axis direction, the tortoise 62 as NPC is turned upside down (a tortoise upside down returns to the normal position). The tortoise in an upside-down position is easy to slide, so that the movement process is made to give a greater tilt-input moving amount than that of the normal position.

10   Figure 16 is a memory map of the program ROM 34. The program ROM 34 stores a game program and game data to be executed by the CPU 21. The program ROM 34 concretely includes an object character data memory area 34a, a map data memory area 34b, an acceleration-sensor output value conversion table memory area 34c and a game program memory area 34e. The object character data memory area 34a stores graphic data of the object characters. Because the object character has some poses (e.g. tortoise

15   "normal position" and "upside-down position", etc.), for each character a plurality of ones of graphic data are stored for a plurality of poses. The map data memory area 34b stores map data on a game map basis and game-map-select maps. The game-map select map is virtual map data to be displayed on the LCD 12 during a game map select process

20   hereinafter described with reference to Figure 30.

The acceleration-sensor output value conversion table memory area 34c stores conversion tables to convert output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32, for utilization in a game program. The conversion tables includes a game map select table, a player character moving table and an NPC moving table. Also,

25   the player character moving table includes tables for in-air, on-floor, on-ice and under-

- 20 -

water, which are to be selected depending upon a land coordinate where a player character is present. The NPC moving table includes tables for normal position and upside-down position. The tortoise as NPC assumes states of normal and backside-down positions, depending upon which a table is to be selected. The details of the tables will be hereinafter described with reference to Figure 20 to Figure 26.

The game program memory area 34e stores varies game programs to be executed by the CPU 21. Specifically, stored are a main program hereinafter described with reference to Figure 27, a 0G set program hereinafter described with reference to Figure 28, a neutral-position set program hereinafter described with reference to Figure 29, a game map select program hereinafter described with reference to Figure 30, a sensor output read program hereinafter described with reference to Figure 31, an object moving program hereinafter described with reference to Figure 32 to Figure 36, a collision program hereinafter described with reference to Figure 37, a screen scroll program hereinafter described with reference to Figure 40, an NPC self-controlled moving program and other programs.

Figure 17 is a memory map of the work RAM 26. The work RAM 26 is to store temporary data for executing a game program by the CPU 21. Specifically, included are a neutral position data memory area 26a, an acceleration sensor memory area 26b, an impact input flag memory area 26c, a map select screen camera coordinate memory area 26e, a game map number memory area 26f and a character data memory area 26g.

The neutral position data memory area 26a stores neutral position data (NPx, NPy, NPz) to be set in a neutral-position set process hereinafter described with reference to Figure 29. This is data concerning a reference tilt of the portable game apparatus for playing a game.

The acceleration-sensor output value memory area 26b stores output values (INx,

- 21 -

INy, INz) of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 which are detected by the acceleration sensor 31 and contact switch 32 and to be read out through the sensor interface 33 in a sensor output read process of Figure 31. The impact input flag memory area 26c stores an impact input flag (FS) that assumes 1 when equal to or greater than a constant value is the magnitude of resultant vector of a vector component in the X-axis direction taken of an acceleration input in the X-axis direction and a vector component in the Y-axis direction taken of an acceleration input in the Y-axis direction. The determination of impact input is executed in a sensor output read process of Figure 31.

The map select screen camera coordinate memory area 26e stores coordinates (Cx, Cy) at upper left corner of an LCD 12 display area in a game map select map which is to be displayed in a game map select process hereinafter described with reference to Figure 30. The game map number memory area 26f stores corresponding number data (MN) to a game map having been selected by a player during a game map select process hereinafter described with reference to Figure 30.

The character data memory area 26g stores, for each of the player characters and NPCs, moving acceleration data (Ax, Ay, Az), moving-acceleration change amount data (dAx, dAy, dAz), velocity data (Vx, Vy, Vz), coordinate data (X, Y, Z), last-time coordinate data (Px, Py, Pz), current position status (SP) and pose numbers (PN).

The coordinate (Px, Py, Pz) in the last time is for returning to the last-time coordinate a player character or NPC when collided with a wall or the like. The current-position status data (SP) is data concerning a land at a coordinate where the player character is present. Based on this data, an acceleration-sensor output value conversion table (in-air, on-floor, on-ice, on-water) is to be selected. The pose number (PN) is data concerning a character state (pose) (e.g. tortoise normal and upside-down positions, etc.).

Figure 18 is a memory map of the display RAM 25. The display RAM 25 is to temporarily store display data obtained through executing a game program by the CPU 21. The display RAM 25 has an object data memory area 25a, a scroll counter data memory area 25b and a map data memory area 25c. The object data memory area 25a stores data of the existing characters in the LCD 12 display area among all the characters to appear in a game. Specifically, stored area X-coordinates, Y-coordinates, character IDs, and pose numbers.

The scroll counter data memory area 25b stores a relative coordinate of an upper left corner of the LCD 12 display area of the game. The map data memory area 25c stores game map data of the game map in an area to be displayed on the LCD 12.

Figure 19 is a memory map of the backup RAM 35. The backup RAM 35 stores 0G position data (ZGx, ZGy) to be set in a 0G set process hereinafter described with reference to Figure 38. The 0G position data is to cope with not to have a sensor output value of 0 because of the error possessed by the XY-axis acceleration sensor even when the portable game apparatus is held horizontal. A sensor output value when the portable game apparatus is held horizontal is stored as 0G position data in the backup RAM 35, which in a game process is subtracted from a sensor output value.

Figure 20 to Figure 26 illustrate in detail conversion tables stored in the acceleration-sensor output value conversion table memory area 34c of the program ROM 34. The tables store data, concerning utilization ways and correction of limiting a maximum values, etc., for utilizing, in game processing, sensor output values (INx, INy, INz) of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 and impact input flag (FS). Specifically, stored is data concerning utilization ways, correction ratio, special correction conditions and special correction numbers. The tables are stored in plurality, including a game-map select process table, player-character moving table and

an NPC moving table.

The game map select processing table shown in Figure 20 is to be made reference to in a game map select process hereinafter described with reference to Figure 30. The output values (INx, INy) of the XY-axis acceleration sensor by this table are utilized for calculating a camera coordinate (Cx, Cy) change amount. Incidentally, because the correction ratio is wise, the camera coordinate (Cx,. Cy) will be moved twice the output value (INx, INy) of the XY-axis acceleration sensor 31. The output value (INz) of the Z-axis contact switch 32 is utilized for a map determining process. The impact input flag (FS) is not utilized.

The player character moving table shown in Figure 21 to Figure 24 is made reference to in a tilt movement process to be executed at step S33, and in an impact movement process to be executed in step S33 in a player character moving process hereinafter described with reference to Figure 33. The player character moving table includes tables for in-air, on-floor, on-ice and under-water, Any one of the conversion tables is to be selected and referred to in accordance with a coordinate topology where the player character is present (current position status).

In the player character moving table, the output value X (INx) of the XY-axis acceleration sensor 31 is utilized for calculating a change amount (dAx) of an X-movement acceleration of a player character while the output value Y (INy) is utilized for calculating a change amount (dAy) of an Y-movement acceleration. In the case the current position status is "in-air", the moving-acceleration change amount (dAx, dAy) is zero by referring to Figure 21. For the case of "on-floor", because the correction ratio if referred to Figure 22 is twice, twice the output value (INx, INy) of the XY-axis acceleration sensor 31 gives a change amount (dAx, dAy) of moving acceleration. Also, where the output value (INx, INy) of the XY-axis acceleration sensor is greater than 20

- 24 -

due to particular correction condition 1, the moving-acceleration change amount (dAx, dAy) is limited to "40". For "on-ice", by referring to Figure 23 three times the output value (INx, INy) of the XY-axis acceleration sensor 31 gives a change amount (dAx, dAy) (greater moving amount "on-ice"). Meanwhile, where the output value (INx, INy) of the XY-axis acceleration sensor is greater than "20" due to particular correction condition 1, the moving-acceleration change amount (dAx, dAy) is limited to "60". For "under-water", by referring to Figure 24 a half of the output value (INx, INy) of the XY-axis acceleration sensor 31 gives a moving-acceleration change amount (dAx, dAy) (smaller moving amount "in water"). Also, where the output value (INx, INy) of the acceleration sensor 31 is greater than "10" due to particular correction condition 1, the change amount (dAx, dAy) is limited to "5".

In the player character moving tables, the output value (INz) of the Z-axis contact switch 32 is utilized to calculate a change amount (dAz) of Z-movement acceleration. There is no special correction condition.

In the player-character moving table, an impact input flag (FS) has an effect upon X and Y moving-acceleration change amounts (dAx, dAy). In the case the present position status is "in-air" and "under-water", the impact input flag (FS) is ignored by referring to Figure 21 and Figure 24. Where the present position status is "on-floor", with reference to Fig. 22 processing is made to multiply by 3 times the X and Y moving-acceleration change amounts (dAx, dAy). Where the current position status is "on-ice", with reference to Figure 23 processing is made to multiply by 5 times the X and Y moving-acceleration change amounts (dAx, dAy). In this manner, when there is an impact input, for "on-floor" and "on-ice" the X and Y moving-acceleration change amounts (dAx, dAy) are increased (moved at higher speed) as compared to the usual.

The NPC moving tables of Figure 25 and Figure 26 are to be referred to in a tilt

movement process in step S44 and impact moving process in step S45 of an NPC moving process hereinafter described with reference to Figure 34. The NPC moving tables includes tables for normal and upside-down positions. Any one of the two conversion tables is selected and referred to depending upon a pose (normal or upside-down) of a tortoise as NPC.

In the NPC moving table, an output value X (INx) of the XY-axis acceleration sensor 31 is utilized to calculate a change amount (dAx) of an NPC X movement acceleration while an output value Y (INy) is utilized to calculate a change amount (dAy) of a Y movement acceleration. For the "normal position", because with reference to Fig. 25 the correction ratio is 1/2 times, 1/2 times an output value (INx, INy) of the XY-axis acceleration sensor 31 gives an X-and-Y moving-acceleration change amount (dAx, dAy). Also, where the output the values (INx, INy) of the XY-axis acceleration sensor 31 is smaller than 10 under special correction condition 1, the moving-acceleration change amount (dAx, dAy) is 0 (in the "normal position", with a small tilt the tortoise will brace its legs and not slide). Also, where the output (INx, INy) of the XY-axis acceleration sensor 31 is greater than 20 under special correction condition 2, the moving-acceleration change amount (dAx, dAy) is limited to 10. For the "upside-down position", with reference to Figure 26, 2 times an output value (INx, INy) of the XY-axis acceleration sensor 31 gives an X-and-Y moving-acceleration change amount (dAx, dAy) (moving amount greater because the tortoise "backside-down" easily slide as compared to "normal"). Also, where the output value (INx, INy) of the XY-axis acceleration sensor 31 is greater than 20 under special correction condition 1, the moving-acceleration change amount (dAx, dAy) is limited to 40.

In the NPC moving tables, the output value (INz) of the Z-axis contact switch 32 is utilized to determine tortoise inversion to a normal or inverted position. Each time the

output value of contact switch 32 becomes "1", the tortoise turns to a normal or inverted

state in a repetitive manner. The impact input flag (FS) is not utilized for the NPC

movement process.

Figure 27 is a flowchart of a main routine. If a cartridge 30 is loaded onto the

game machine main body 10 and the power switch of the game machine main body 10 is

turned on, the CPU 21 starts to process the main routine of Figure 33. First, in step S11 it

is determined whether it is a first starting or not, or whether a player requested for 0G

setting (e.g. whether started while pressing the operation key 13b of Figure 1) or not. If

not a first starting and there was no 0G set request, the process advances to step S13.

Meanwhile, when a first starting or there was a 0G set request, a 0G set process

hereinafter described with reference to Figure 28 is made in step S12 and then the process

proceeds to step S14. In the step S14, a neutral-position set process hereinafter described

with reference to Figure 29 is made and then the process advances to step S17. Here, the

neutral-position setting is meant to set a reference tilt of the portable game apparatus for

playing a game. The recommended position setting is meant to set a neutral position

based on data wherein the data is concerned with a proper neutral position in accordance

with a game content (the recommended position sight target coordinate 34d of the

program ROM 34) that have been previously memorized in a game program.

In step S17 a game map select process hereinafter described with reference to

Figure 30 is performed so that one of a plurality of game maps is selected by the player.

After the step S17, the process advances to a main loop.

The main loop is a process of from step S19 to step S29, which is repeatedly

executed until game over or game clear is reached. In step S19, required data is written to

the display RAM 25 based on a coordinate (X, Y, Z) and pose number (PN) of the

character data 26g of the work RAM 26, object character data 34a of the program ROM

34 and map data 34b. Based on the data stored in the display RAM, a game scene is displayed on the LCD 12. In step S20 a sensor output read process hereinafter described with reference to Figure 31 is performed. The output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 are read out through the sensor interface 33 and then corrected. After the step S20, in step S21 it is determined whether there was a neutral-position set request or not. If there was no request, the process advances to step S23 while if there was a request the process proceeds to step S22 to perform a neutral-position set process. After resetting a neutral position, the process returns to step S19. This means that one operation switch (e.g. operation switch 13e shown in Figure 1) is assigned to an exclusive operation switch for neutral-position setting so that neutral-position setting can be made at any time by pressing the operation switch 13e even during playing a game.

In step S23 it is determined whether the impact input flag is ON or not. If the impact input flag is OFF, the process proceeds to step S26 while if ON the process advances to step S24 to determine whether the topology of current coordinate that the player character is present is under-water or not (determined based on a current position status). If not under-water is determined, the process advances to step S26 while if determined under-water, the process advances to step S25 to perform a wave producing process (display is as shown in the middle portion in Figure 15). Specifically, processing is made to cause waves in a direction and with a magnitude depending on a resultant vector, wherein the resultant vector is given by a vector component in the X-axis direction taken of a sensor output value X (INx) and a vector component in the Y-axis direction is taken of a sensor output value Y (INy). The player can have a feeling as if the impact applied by him or her to the portable game apparatus was reflected in an environment (water) of the game space. After step S25, the process proceeds to step S26.

In the step S26 an each-character moving process hereinafter described with reference to Figure 32 to Figure 35 is performed thereby performing a process of moving the player character and NPC. After the step S27, a collision process hereinafter described with reference to Figure 37 is performed thereby performing a process of colliding the player character with NPC, etc. After the step S27, a scroll process hereinafter described with reference to Figure 40 is performed.

Figure 28 shows a subroutine flowchart for a 0G set process. This subroutine performs a process to store as 0G position data to backup RAM 35 an output value of the XY-axis acceleration sensor 31 when the portable game apparatus (specifically, the LCD 12 display surface) is held horizontal.

In step S121 "POSITION HORIZONTAL TO GROUND AND PRESS OPERATION SWITCH" is displayed on the LCD 12, requesting the player to hold the portable game apparatus (specifically, the LCD 12 display surface) in a horizontal state. In step S122 an operation switch input process is performed. In step S123, if depression of an operation switch (e.g. operation switch 13b of Figure 1) for determination is determined, it is then determined in step S124 whether the Z-axis contact switch 32 is ON or not. When the Z-axis contact switch 32 is ON, an alert sound is generated in step S125 and the process returns to step S121. This is because, where the Z-axis contact switch is ON, the LCD in its display surface is directed downward and the player is requested to perform setting again. In step S124, where the Z-axis contact switch is determined OFF, then in step S126 the output value of the XY-axis acceleration sensor 31 at this time is stored as 0G position data to the backup RAM 35.

Figure 29 is a subroutine flowchart for a neutral-position set process. This subroutine performs process that the player arbitrarily determines a portable game apparatus at a holding angle easy to play a game. The output value of the XY-axis

- 29 -

acceleration sensor 31 and Z-axis contact switch 32 at that time are stored as neutral position data to the work RAM 26.

In step S141 "POSITION AT ANGLE EASY TO PLAY AND PRESS OPERATION SWITCH" is displayed on the LCD 12. In step S142 an operation switch input process is made. In step S143, if the depression of an operation switch for determination (e.g. operation switch 13b of Figure 1) is determined, then in step S144 correction is performed by subtracting 0G position data from an output value of the XY-axis acceleration sensor 31 at this time (the neutral position data is rendered as data corresponding to a tilt with respect to the horizontal state). Then, in step S145 a correction value of the output of the XY-axis acceleration sensor (calculation result of step S144) and an output value of the Z-axis contact switch 32 are stored as neutral position data to the neutral position data memory area 26a of the work RAM 26.

Figure 30 is a flowchart of a game map select process. In this subroutine, the player selects any one of a plurality of game maps stored in the game program. The screen of game map select process is displayed, for example, as shown in Figure 13 mentioned before. On the LCD 12, one area of a game-map select map is displayed. The player makes slide input in the X-axis or Y-axis direction to move the display area on the LCD 12 thereby displaying map icons (A, B, C, D in Figure 16) within the display area. Then, a movement is inputted in the Z-axis direction. This results in selection of a game course corresponding to a course icon being displayed on the LCD 12 upon inputting the movement (or impact) in the Z-axis direction.

First, in step S171 a camera coordinate (Cx, Cy) is initialized. Then, in step S172 one area of the game-map select map is displayed on the LCD 12 based on the camera coordinate (Cx, Cy). In step S173 a sensor output read process hereinafter described with referring to Figure 31 is made. As a result, the output values of the XY-axis acceleration

sensor 31 and Y-axis contact switch 32 are read out and corrected. In step S174 a table

shown in Figure 26 is referred to. Specifically, the camera coordinate (Cx, Cy) is

changed based on the sensor output values (INx, INy). Specifically, because the

correction ratio is twice, the camera coordinate (Cx, Cy) is varied by an amount twice the

5   sensor output value (INx, INy). For example, when the sensor output value (INx) is 5, the

camera coordinate (Cx) is rendered +10. In step S175 it is determined whether the

display area based on the camera ordinate (Cx, Cy) is outside a range of the game map

select map or not. If not outside the range, the process advances to step S177 while if in

outside the range the process proceeds to step S176. In step S176 correction is made so as

10   to display an end area of the game-map select map and then the process proceeds to step

S177. In the step S177 it is determined whether the Z-axis contact switch 32 is ON or not.

If the contact switch 32 is determined OFF, the process returns to step S172. If the Z-axis

contact switch 32 is determined ON, then it is determined in step S178 whether any one of

the map icons (A, B, C, D in Figure 16) is displayed in the display range of the LCD 12 or

15   not. If it is determined that no map icon is displayed within the display area, then in step

S179 an alert sound is generated and the process returned to step S172. If it is determined

that a map icon is displayed within the display range, then in step S181 a corresponding

game map number (MN) to the map icon being displayed is stored to the work RAM 26.

Figure 31 is a flowchart for a sensor output read process. In this subroutine, the

20   output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 are read

out and corrected. Specifically, from the data of the latch 334 and latch 335 of the sensor

interface 33 are read output values (INx, INy) of the acceleration sensor and an output

value (INz) of the Z-axis contact switch 32. Furthermore, a correction process is made

based on 0G position data and neutral position data.

25   In step S201, data is read out of the latch 334 and latch 335. In step S202,

acceleration-sensor output values (INx, INy) and Z-axis contact switch output value (INz) are read from the latch data, and stored to the acceleration-sensor output value memory area 26b of the work RAM 26. In step S203 it is determined whether there was an impact input or not. Specifically, it is determined whether equal to or greater than a

5   given value a magnitude of a resultant vector having vector component in the X-axis direction taken of the acceleration sensor 31 output value X (INx) and a vector component in the Y-axis direction taken of the acceleration sensor 31 output value Y (INy). If determined equal to or greater than a given value, then in step S204 the impact input flag (FS) is set "ON" and the process advances to step S206. If the resultant vector

10   magnitude is determined smaller than the given value, then in step S205 the impact input flag (FS) is set "OFF" and the process advances to step S206. In step S202, processing is made to subtract the 0G position data memorized in the backup RAM 35 from the data of the acceleration-sensor output value memory area 26b. In step S207, the value further corrected with the neutral position data is stored as INx, INy and INz to the

15   acceleration-sensor output memory area 26b.

The correction with the neutral position data is performed, specifically, on the output value X (INx) and output value Y (INy) of the acceleration sensor by subtracting the values of the neutral position data (NPx, NPy). For the output value (INz) of the Z-axis contact switch 32, when the value of neutral position data (NPz) is "1", processing is

20   made to invert "0" and "1".

Figure 32 to Figure 36 are flowcharts for an object moving process. Figure 32 is an object moving process main routine flowchart. In step S261, a player-character moving process is performed that is hereinafter described with reference to Figure 33. In step S262, an NPC moving process is performed that is hereinafter described with

25   reference to Figure 34. The NPC moving process is repeated the number of NPCs.

Figure 33 is a player-character moving process flowchart. In step S31, a present coordinate (X, Y, Z) of the player character is stored by copy as a last-time coordinate (Px, Py, Pz). This is required to return the player character collided with a wall to a last-time coordinate, in a collision process hereinafter described with reference to Figure 37. In step S32, a moving-acceleration change amount (dAx, dAy, dAz) is initialized, and then in step S33 a tilt movement process is performed. In the tilt movement process, reference is made to proper one of the conversion tables shown in Figure 21 to Figure 24 depending upon a present position status of the player character, to make processing of calculating an X-and-Y moving-acceleration change amount of the player character. This processing determines a moving-acceleration change amount (dAx, dAy) such that the character is rolled (slid) responsive to a tilt (tilt input) of the portable game apparatus. Furthermore, in step S34, an impact moving process is performed. In the impact moving process, reference is made to proper one of the conversion tables of Figure 21 to Figure 24, to make processing of increasing an X-and-Y change amount of the player character. This process increases a moving-acceleration change amount (dAx, dAy) such that the player character makes a dash (moves at higher speed) when applying an impact input. In step S35, a jump moving process is made that is hereinafter described with reference to Figure 35. After the step S35, it is determined in step S36 whether a wave generation process in step S25 of the flowchart of Figure 27 has been made or not. If no wave generation is determined, the process advances to step S38. If waves have generated is determined, in step S37 a wave moving process hereinafter described with reference to Figure 36 is made, and then the process proceeds to step S38. In the step S38, a moving acceleration (Ax, Ay, Az) is calculated based on the moving-acceleration change amount (dAx, dAy, dAz) calculated in the tilt moving process, impact moving process, jump process and wave moving process of the steps S33 to S37, and a velocity (Vx, Vy, Vz) is calculated

based on the moving acceleration (Ax, Ay, Az). In step S39, a coordinate (X, Y, Z) is calculated based on the velocity (Vx, Vy, Vz).

Figure 34 is a flowchart of an NPC movement process. In step S41 a current coordinate (X, Y, Z) is stored by copy to the last-time coordinate (Px, Py, Pz). In step S42 the moving-acceleration change amount (dAx, dAy, dAz) are initialized. In step S43 an NPC self-controlled movement process is executed based on the game program. Specifically, a moving-acceleration change amount (dAx, dAy, dAz) e.g. for a tortoise is determined based on a random number value. After the step S43, in step S44, a tilt movement process is executed. In the tilt movement process, processing is made to calculate an NPC X-and-Y moving-acceleration change amount by referring to a suited one of the conversion tables shown in Figure 25 or Figure 26 according to an NPC pose number. Furthermore, in step S45 an impact process is made. However, in the present embodiment, the NPC will not be affected by impact input. In step S46 it is determined whether a wave producing process has been made in step S25 of the flowchart of Figure 25 or not. If no wave production is determined, the process advances to step S48. If waves have been produced is determined, then in step S47 a wave movement process hereinafter described with reference to Figure 36 is executed and then the process advances to step S48.

In step S48, a moving acceleration (Ax, Ay, Az) is calculated based on the moving-acceleration change amounts (dAx, dAy, dAz) determined by the self-controlled movement process, tilt movement process, impact movement process and wave movement process of steps S43 to S47. Furthermore, a velocity (Vx, Vy, Vz) is calculated based on the movement acceleration (Ax, Ay, Az). In step S49 a coordinate position (X, Y, Z) is calculated based on the velocity (Vx, Vy, Vz). In step S51 it is determined whether an output value (INz) of the Z-axis contact switch is "1" or not. In

- 34 -

the case that the Z-axis contact switch output value (INz) is "0", the NPC movement

process subroutine is ended. Where the Z-axis contact switch output value (INz) is "1",

an inversion process to a normal or upside-down position is executed in step S52.

Specifically, a pose number (PN) of the character data in the work RAM 26 is changed.

5   Figure 35 shows a flowchart of a jump process. In this subroutine, when there is a

movement input in the Z-axis direction, processing is made to cause the player character

to jump. Also, when there is no movement input in the Z-axis direction in a state the

player character is in the air, processing is made to descend the player character.

In step S351 it is determined whether the output value (INz) of the Z-axis contact

10   switch 32 is 1 or not. When the output value (INz) of contact switch 32 is "1", the current

position status (PS) is set as "in-air" in step S52. Thereafter in step S353 the Z moving-

acceleration change amount (dAz) is rendered "1". When the output value (INz) of the

Z-axis contact switch 32 is "0" in the step S351, it is determined in step S354 whether the

player character is "in-air" or not. When not "in-air", the jump process is ended. Where

15   "in-air" in the step S354, the Z moving-acceleration change amount (dAz) is rendered

"−1" in step S355 and then the jump process is ended.

Figure 36 shows a flowchart of a wave movement process. In this subroutine,

processing is made to calculate a moving-acceleration change amount due to the waves

produced due to impact input by the player. In step S361 a current position status is read

20   in. In step S362 it is determined whether the current position status is in a position to

undergo an affection of waves or not (i.e. "under-water" or not). If determined as a

position free from an affection of waves, the wave movement process is ended. If

determined as a position to undergo an affection of waves, then in step S363 are

calculated respective X and Y moving-acceleration change amounts due to an affection of

25   waves and added to the X and Y moving-acceleration change amounts calculated by the

tilt movement process and impact movement process.

Figure 37 shows a flowchart of a collision process. In steps S271 to S275, an NPC collision determination process is carried out. The NPC collision determination process is repeated to the number of NPCs. In step S271 it is determined whether an NPC has collided with a wall or not. If determined as collision with a wall, the process proceeds to step S273. If no collision is determined, the process advances to step S272 wherein it is determined whether there has been a collision with another NPC or not. If determined as collision with another NPC, the process advances to step S272. If determined as no collision with another NPC, the process proceeds to step S273. Where determined as a collision with a wall or another NPC, then in step S273 an impact sound is generated and then in step S274 the NPC coordinate (X, Y, Z) is returned to the last-time coordinate (Px, Py, Pz), and the process advances to the step S275.

In step S275, a current position status of NPC is detected and stored in the work RAM 26. After step S275 it is determined in step S276 whether the player character has collided with a wall or not. If no collision against wall is determined, the process proceeds to step S279. If a collision with a wall is determined, then in step S277 an impact sound is generated and then in step S278 the player character coordinate (X, Y, Z) is returned to the last-time coordinate (Px, Py, Pz), and the process advances to step S279.

In step S279, a current position status of the player character is detected and stored in the work RAM 26. After step S279, it is determined in step S281 whether the player character has collided with an NPC or not. If a collision against an NPC is determined, a process is made in step S282 to vanish the NPC. After step S282, it is determined in step S283 whether all the NPCs have been vanished or not. If all the NPCs have vanished is determined, a game clear process is executed in step S284. When no collision with an NPC is determined in step S281 or when all the NPCs have not vanished is determined in

step S283, the process proceeds to step S285. In step S285 it is determined whether the player character has fallen in a hole or not. If determined fallen in a hole, a game over process is effected in step S286. Where the determination is not fallen in a hole, the impact process is ended.

5   Figure 38 and 39 each show one example of a scene showing on-screen scroll. In the scene, there are displayed a ball as a player character, tortoises 62a – 62c as NPC, and a wall 63 and hole 64 forming a maze. The dotted lines 65 show a limit of screen scroll (actually, the dotted lines 65 will not be displayed on the LCD 12). The game map is a virtual map that is broader than LCD 12 display area, as stated before. On the LCD 12 is

10   displayed part of a game map around the player character 61. When the player tilts or so the portable game apparatus and the player character 61 is moving to an outer area of the dotted lines 65, the scene is scrolled moving the game-map display area over the LCD12. Furthermore, the player character 61 and NPC 62 are moved to and displayed in a position toward a center of a scene by a corresponding amount to scrolling. In this

15   manner, screen scrolling makes possible game play with a broader game map. For example, if the player character is going beyond the dotted line 65 to a left side area as shown in Figure 38, the game map area in display is scrolled to left so that the player character 61 and NPC can be moved to and displayed in a position by a corresponding amount to scrolling (Figure 39). Note that the scroll rate may be changed depending upon

20   a magnitude of tilt input.

Figure 40 shows a flowchart of a screen scroll process. In step S291 it is determined whether the player character is out of a scroll area in an X-axis minus direction or not. Here, the scroll area refers to an area as surrounded by the dotted lines 65 shown in Figure 38. If determined not out of the area with respect to the X-axis minus

25   direction, the process advances to step S294. If determined out of the area in the X-axis

minus direction, it is then determined in step S292 whether the current display area on the LCD 12 is a left end area of the game map or not. If determined as a left end area, the process advances to step S294. If determined not a left end area, then in step S293 a scroll counter X coordinate (SCx) stored in the display RAM 25 is decreased by a given amount and then the process proceeds to step S294. In step S294 it is determined whether the player character is out of the scroll area with respect to the X-axis plus direction or not. When determined not out of the area in the X-axis plus direction, the process advances to step S297. When determined out of the area in the X-axis plus direction, it is determined in step S295 whether the current display area on the LCD 12 is a right end area of the game map or not. If determined as a right end area, the process advances to step S297. When determined not a right end area, in step S296 the scroll counter X coordinate (SCx) is increased by a given amount and then the process proceeds to step S297.

In step S297 it is determined whether the player character is out of the scroll area in a Y-axis minus direction or not. If determined not out of the area in the Y-axis minus direction, the process advances to step S301. When determined out of the area in the Y-axis minus direction, it is determined in step S298 whether the current display area on the LCD 12 is an upper end area of the game map or not. If determined as an upper end area, the process proceeds to step S301. When determined not an upper end area, in step S299 a scroll counter Y coordinate (SCy) is decreased by a given amount and then the process proceeds to step S301. In step S301 it is determined whether the player character is out of the scroll area in a Y-axis plus direction or not. When determined not out of the area in the Y-axis plus direction, the screen scroll process is ended. When determined out of the area in the Y-axis plus direction, it is determined in step S302 whether the current display area on the LCD 12 is an lower end area of the game map. When determined as a lower end area, the screen scroll process is ended. When determined not a lower end area, in

step S303 the scroll counter Y coordinate (SCy) is decreased by a given amount and then the screen scroll process is ended.

(Second Embodiment)

Next, a portable game apparatus according to a second embodiment of the invention will be explained with reference to Figure 41 to Figure 49. The second embodiment is common in external view, XY-axis definition diagram, block diagram, sensor-interface measurement principle diagram and Z-axis contact switch structural view to Figure 1 to Figure 7 of the first embodiment, hence omitting explanations thereof.

Figure 41 illustrates an example of a game scene in the present embodiment. In this game, a player can give impact to the portable game apparatus to cause an upheaval in a game-space land, enjoying the game while controlling the movement of a game character.

As shown in Figure 41(a), a game-character tortoise 81 and a land-upheaval character 82 are displayed in a game scene. As shown in Figure 41(b), the tortoise 81 is moved self-controlled according to a game program. In a state shown in Figure 41(b), when an impact input is given in the Z-axis direction to the portable game apparatus, the land-upheaval character 82 is displayed higher and greater with upheaval, as shown in Figure 41(c). This controls the tortoise 81 to slide (tortoise 82 in advancing retracts due to land upheaval). By thus processing, it is possible to provide the player with a feeling as if the game-space land receives energy and is upheaved when an impact is applied in the Z-axis direction to the portable game apparatus.

Figure 42 is one example of a game scene illustrating a land-upheaval process due to an impact input in the Z-axis direction. In Figure 42(a), an outer frame 12' designates a whole game space and an inner frame 12 a display area to be displayed on the LCD 12.

The game space is a world greater than a display area of the LCD 12. The LCD 12 displays a part of the game space. In the game space, there are twelve land-upheaval characters 82 (82a - 82l) and three tortoise characters 81 (81a -81c). Among them, four land-upheaval characters (82a, 82b, 82e, 82f) and one tortoise character (82a) are being displayed on the LCD 12.

In a state shown in Figure 42(a), if an impact input is applied in the Z-axis direction to the portable game apparatus, the twelve land-upheaval characters (82a - 82l, the land-upheaval characters all over the game space) are raised by one step and displayed higher and greater, as shown in Fig. 42(b). At this time, the tortoise characters (81a and 81b) existing at land upheaval are displayed sliding due to the upheaval of land.

In a state shown in Figure 42(b), when an impact input is applied in the Z-axis direction while operating the button A (operation switch 13b), only the four land-upheaval characters (82a, 82b, 82e, 82f) being displayed on the LCD 12 are further raised by one step and displayed higher and greater. In also this case, the tortoise character (81a) existing at land upheaval is displayed sliding due to the land upheaval. By thus processing, when applying an impact input in the Z-axis direction while pressing the button A, it is possible to provide the player with a feeling as if energy due to impact was given to the game space limited to the area being displayed on the LCD 12.

Incidentally, although not shown, if in the state shown in Figure 42(b), an impact input is given in the Z-axis direction while operating the button B (operation switch 13c), only the eight land-upheaval characters (82c, 82d, 82g, 82h, 82i - 82l) not being displayed on the LCD 12 are raised by one step and displayed higher and greater. In also this case, the tortoise characters (81b, 81c) existing at the land upheaval are displayed sliding due to the land upheaval. By thus processing, where an impact input is given in the Z-axis direction while pressing the button B, it is possible to provide the player with a feeling as

if energy due to impact was supplied to the game space limited to the area not being displayed on the LCD 12.

Figure 43 is one example of a game scene illustrating a scroll process for a game space on display. The game space on display is to be scrolled by giving a slide-input to the portable game apparatus (see Figure 9 in the first embodiment). For example, in Figure 43(a) land-upheaval characters 82a, 82b, 82e, 82f and tortoise character 81a are displayed on the LCD 12. In this state, when the portable game apparatus is slid in a Y-axis minus direction, the game space on display is scrolled down, resulting in display of land characters 82e, 82f and tortoise character 81a as shown in Figure 43(b).

Also, in a state shown in Figure 43(b), when the portable game apparatus is slid in an X-axis plus direction, the game space on display is scrolled right, to provide display with a land character 82f and tortoise character 81a. By thus processing, it is possible for the player to enjoy a game with a game space greater than the LCD 12. Also, because as stated before an effect (land upheaval) can be given to the game space limited to an inside or outside of the area of display by the use of the button A or button B, the player can enjoy a complicated game.

Figure 44 illustrates control of scenes with temperature increase caused by impact input in XY-axis directions. Although the tortoise characters 81a - 81c moves in a self-controlled fashion according to the game program as stated before, this self-controlled movement becomes more active as temperature increases (specifically, moving amount increases). In a state shown in Figure 44(a), when an impact input is applied in the XY-axis direction (see Figure 11 in the first embodiment), a parameter of temperature increases to provides display that the tortoise characters 81a - 81c are actively moving. By thus processing, it is possible to provide the player with a feeling as if energy was supplied and the temperature was increased in the game space upon giving an impact in

the XY-axis direction to the portable game apparatus.

Hereunder, explanations will be made on the data stored on the memory with reference to Figure 45 and Figure 46.

Figure 45 is a memory map of the program ROM 34. The program ROM 34 stores

5　a game program and game data to be executed by the CPU 21. The program ROM 34, concretely, includes an object-character data memory area 342a, a map-data memory area 342b, a land-upheaval-point data memory area 342c, a scroll-limit value data memory area 342d, an acceleration-sensor output value conversion table memory area 342e and a game program memory area 342f. The object-character data memory area 342a and the

10　map-data memory area 342b store object characters and game-map graphic data. The land-upheaval-point data memory area 342c stores position data (X coordinate and Y coordinate; Px1 - Px12, Py1- Py12) in a game space for each of the land upheaval characters (82a - 82l) shown in Figure 42. The scroll-limit-value data memory area 342d stores data representative of scroll limit values (SCxmax, SCymax) in order not to make

15　scrolling at an up, down, left or right end of the game space when scrolling the game space.

The acceleration-sensor output value conversion table memory area 342d stores a conversion table to convert, and utilize in a game program, output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32. Specifically, stored is data similar to

20　that of the conversion tables (Figure 20 to Figure 26) of the first embodiment. It is defined that a sensor output value X (INx) and sensor output value Y (INy) is to be utilized in calculating a change amount of a scroll counter X coordinate (SCx) and Y coordinate (SCy) in a range-of-sight moving process hereinafter described with reference to Figure 48. Due to this, by giving a slide-input to the portable game apparatus (see

25　Figure 9 in the first embodiment), the game space on display is scrolled thereby making

processing to move the range of sight. Also, definition is made to utilize a Z-axis contact switch output value (INz) in land upheaval determination. Definition is made to utilize an impact input flag (FS) in temperature rise determination.

The game program memory area 342f stores a game program to be executed by the CPU 21. Specifically, stored are a main program hereinafter described with reference to Figure 47, a sensor output read program similar to Figure 31 of the first embodiment, a range-of-sight moving program hereinafter described with reference to Figure 48, a land upheaval program hereinafter described with reference to Figure 49, a temperature raising program, a tortoise-character control program and other programs.

Figure 46 is a memory map of the work RAM 26. The work RAM 26 stores temporary data for the CPU 21 to execute a game program. Specifically, included are an acceleration-sensor output value memory area 162a, an impact input flag memory area 262b, a land-upheaval data memory area 262c, a temperature data memory area 262d and a character data memory area 262e.

The data stored on the acceleration-sensor output value memory area 262a and impact input flag memory area 262b is similar to that of the first embodiment, hence omitting explanations. The land-upheaval data memory area 262c stores height data concerning respective points of land upheaval. The height data is varied according to an impact input in the Z-axis direction in a land upheaval process hereinafter described with reference to Figure 49. Based on this data, the land upheaval characters at respective land upheaval points are determined in state of display. For example, where the height data is 1, the land upheaval character is displayed as shown at 82a in Figure 42(a). Where the height data is 2, display is as shown at 82a in Figure 42(b). Where the height data is 3, display is as shown at 82a in Figure 42(c).

The temperature data memory area stores temperature data for the game space.

The temperature data is varied according to an impact input in the XY-axis direction, in a temperature increase process (in step S64 of the main program shown in Figure 47). This data has an effect upon a tortoise-character control process (self-control movement, in step S65 of the main program shown in Figure 47).

The character-data memory area 262e stores coordinate data (X, Y, Z) and last-time coordinate data (Px, Py, Pz), in the number of the tortoise characters.

The memory map of the display RAM is similar to that of Figure 18 of the first embodiment, hence omitting explanation.

Hereunder, a process flow of a game program will be explained with reference to Figure 47 to Figure 49.

Figure 47 is a main routine flowchart. When a cartridge 30 is inserted to the portable game apparatus main body 10 and the power to the portable game apparatus main body 10 is turned on, a main routine as shown in Figure 47 is started. Although in also the second embodiment a 0G position process or neutral-position set process may be made similarly to the first embodiment, explanation is omitted herein for the sake of simplifying explanation.

First, in step S61 a sensor output read process is carried out similarly to Figure 31 of the first embodiment. This reads output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 through the sensor interface 33 (corrections by 0G position data and neutral position data is omitted). After the step S61, in step S62 a range-of-sight moving process (scroll process of a game space on display) is made that is hereinafter described with reference to Figure 48. After the step S62, in step S63 a land upheaval process is made that is hereinafter described with reference to Figure 49. After the step S63, in step S64 a temperature increase process is made. In the temperature increase process, it is first determined whether there is an impact input in the XY-axis direction or

not. In the case of the presence of an impact input in the XY-axis direction, processing is made to increase a temperature parameter (T) by 1. After the step S64, in step S65 a tortoise-character control process is made. In the tortoise-character control process, a tortoise-character moving process is first made due to self-controlled movement.

5 Specifically, processing is made to calculate a tortoise-character moving amount, e.g. using random values. Incidentally, control is made such that the self-controlled movement of a tortoise character increases in amount as the temperature (T) is higher. Thereafter, a tortoise-character moving process is made with land upheaval. Specifically, processing is made to move the tortoise character sliding when a land under the tortoise

10 character is raised. Incidentally, the tortoise-character control process is repeated the number of the tortoise characters.

After the step S65, in step S66 game-space scrolling as well as display process for a land upheaval object and tortoise character are made based on a result of the range-of-sight moving process, land-upheaval process and tortoise-character control process.

15 Incidentally, where a land upheaval point is raised in height due to the land upheaval process, it would be effective to display the land upheaval character higher and greater together with generation of such sound as imagining an upheaval of a land. After the step S66, it is determined in step S67 whether game is over or not. For example, game-over determination is to be made under a proper condition suited for a game content, including

20 effecting game over, e.g. when a predetermined time has elapsed. If game over is determined in the step S67, the main routine is ended. If no game over is determined in the step S67, the process returns to the step S61.

Figure 48 is a range-of-sight moving process flowchart. First, in step S621 reference is made to conversion table, to perform a process of changing a scroll-counter X

25 coordinate (SCx) and Y coordinate (SCy). After the step S621, it is determined in steps

S622 - S629 whether scroll is about to exceed an end of the game space or not. When scroll is about to exceed a game-space end, processing is made to bring the scroll counter value (SCx, SCy) to a proper value.

In step S622, it is determined whether the scroll-counter X coordinate (SCx) is in excess of a scroll limit value X coordinate (SCxmax) or not. If not in excess of it, the process advances to step S624. When in excess of that is determined in step S622, the process proceeds to step S623. After setting the scroll-counter X coordinate (SCx) value to the scroll limit value X coordinate (SCxmax), the process advances to step S624.

In step S624, it is determined whether the scroll-counter X coordinate (SCx) is smaller than 0 or not. If determined 0 or greater, the process advances to step S626. Where determined smaller than 0 in the step S624, the process proceeds to step S625 to set the scroll-counter X coordinate (SCx) value at 0, and then the process proceeds to step S626.

In step S626, it is determined whether the scroll-counter Y coordinate (SCy) is in excess of the scroll-limit-value Y coordinate (SCymax) or not. If determined not in excess thereof, the process proceeds to step S628. Where determined in excess thereof in the step S626, the process advances to step S627 to set a Y coordinate (SCy) value to the scroll-limit-value Y coordinate (SCymax), and then the process advances to step S628.

In the step S628, it is determined whether the scroll-counter Y coordinate (SCy) is smaller than 0 or not. If determined 0 or greater, the range-of-sight moving process is ended. If determined smaller than 0 in the step S628, the process proceeds to step S629 to set the scroll-counter Y coordinate (SCy) value at 0, and then the range-of-sight moving process is ended.

Figure 49 is a land-upheaval process flowchart. First, it is determined in step S631 whether there is an output of the Z-axis contact switch or not (i.e. whether there is an

impact input in the Z-axis direction or not). Where determined as an absence of a Z-axis contact switch output, the land-upheaval process is ended. Where determined as a presence of a Z-axis contact switch output, the process advances to step S632. In the step S632, it is determined whether the button A (operation switch 13b) is being pressed or not. Where determined that the button A being pressed, the process advances to step S633 to make processing of increasing by 1 the respective land-upheaval points in an area being displayed on the LCD. After the step S633, the land-upheaval process is ended.

If it is determined in the step S632 that the button A is not being pressed, the process proceeds to step S634 to determine whether the button B (operation switch 13c) is being pressed or not. If determined that the button B is being pressed, the process proceeds to step S635 to make processing of increasing by 1 the height (H) of the land upheaval points outside the area being displayed on the LCD. After the step S635, the land upheaval process is ended. If it is determined in the step S634 that the button B is not being depressed, in step S636 all the land upheaval points in height (H) are increased by 1, and then the land upheaval process is ended.

(Third Embodiment)

Next, a third embodiment of the invention will be explained with reference to Figure 50 to Figure 59. This game is to enjoy virtual cooking while moving the portable game apparatus as if it was a frypan or kitchen knife.

Figure 50 to Figure 53 shows examples of game scenes. In Figure 50, in the game scene are displayed a player character 91, a kitchen 92, a cooking stove 93, a frypan 94, a desk 95 and a chopping board 96. When pressing the button A (operation switch 13b), a frypan space process is started that is hereinafter described with reference to Figure 51 and Figure 52. Also, when pressing the button B (operation switch 13c), a kitchen-knife

space process is started that is hereinafter described with reference to Figure 53.

Figure 51 and Figure 52 are examples of game scenes in the frypan space process. In the frypan space process, the portable game apparatus is operated just like a frypan to play a game of cooking a fried egg. In Figure 51(a), a frypan 94 and egg 97 is displayed in the game scene. In a state shown in Figure 51(a), when the portable game apparatus is tilted in a minus direction about the Y-axis, the egg 97 is displayed moving toward left of the frypan as shown in Figure 51(b). Also, in a state shown in Figure 51(b), when the portable game apparatus is tilted in the plus direction about the X-axis, the egg 97 is displayed moving toward the down of the frypan. By thus processing, it is possible to provide the player with a feeling as if he or she operates the portable game apparatus just like a frypan to move an egg by the tilt of the frypan.

In a state shown in Figure 52(a), when an impact input in the Z-axis direction is applied to the portable game apparatus, the egg 97 is displayed jumping above the frypan 94 as shown in Figure 52(b). Thereafter, the egg 97 is displayed landing as shown in Figure 52(c) or (d). At this time, where the egg 97 at an impact input in the Z-axis direction is positioned close to an end of the frypan 94 as shown in Figure 52(a), the egg 97 jumps and lands out of the frypan 94 (Figure 52(c)) thus resulting in failure. Incidentally, in a state shown in Figure 52(b), it is possible to modify a relative positional relationship between the egg 97 and the frypan 94 to land the egg 97 in the frypan 94 by sliding the portable game apparatus (Figure 52(d)). By thus processing, it is possible to provide the player with a feeling as if the portable game apparatus was operated just like a frypan to receive the jumped egg by the frypan.

Figure 53 is examples of game scenes in a kitchen-knife space process. In the kitchen-knife space process, the portable game apparatus is operated just like a kitchen knife to play a game of cutting a cabbage into fine strips. In Figure 53(a), a kitchen knife

- 48 -

98 and cabbage 99 is displayed in the game scene. In the a shown in Figure 53(a), when the portable game apparatus is slid in the plus direction of the X-axis, the cabbage 99 is displayed moving left relative to the kitchen knife 98 as shown in Figure 53(b) (because the kitchen knife 98 is always displayed at a center of the game scene, the cabbage 99 is displayed moving relatively left). By thus processing, it is possible to provide the player with a feeling as if he or she adjusts a position to cut the cabbage by controlling the positional relationship between the cabbage and the kitchen knife.

Furthermore, in the state shown in Figure 53(b), when the portable game apparatus is vertically moved (movement input in the Z-axis direction), the cabbage 99 is displayed being cut by the kitchen knife 98 into fine strips. On this occasion, it will be more effective if generating sound of cutting the cabbage.

Hereunder, explanation will be made on the data stored on the memory with reference to Figure 54. Incidentally, the program ROM 34 stores a program almost similar to the program ROM of the first embodiment (Figure 16). However, the acceleration-sensor output value conversion table memory area stores a table for a frypan, a table for jumping an egg and a table for a kitchen knife. The game program memory area stores main program, a sensor output read program, a frypan space program, an egg jump program, a kitchen knife space program and other programs. Incidentally, the frypan table in the acceleration-sensor output value conversion table will be referred to a frypan space program hereinafter described with reference to Figure 56. The egg-jumping table will be referred to in an egg-jumping program hereinafter described with reference to Figure 58. The kitchen-knife table will be referred to in a kitchen-knife space program hereinafter described with reference to Figure 57.

In the frypan table, the output value (INx, INy) of the XY-axis acceleration sensor 31 is defined to be utilized in calculating a change amount of an egg X-and-Y coordinate

(Ex, Ey). Due to this, the display position of an egg is varied when a tilt is input to the portable game apparatus (see Figure 10 in the first embodiment), thereby displaying and controlling the egg as if it slides over the frypan. Also, the output value (INz) of the coordinate Z-axis contact switch 32 is to be utilized in jump determination of an egg. The impact input flag (FS) is defined not to be utilized.

In the egg jumping table, the output value (INx, INy) of the XY-axis acceleration sensor 31 is defined to be utilized in calculating a change amount of an egg X-and-Y coordinate (Ex, Ey). Due to this, the display position of an egg is varied when inputting a slide to the portable game apparatus while the egg is in jump (see Figure 9 in the first embodiment). This provides display and control as if the relative position of the frypan and the egg was varied. Incidentally, in the egg jumping table, the correction ratio is defined a minus value. This is because in the present embodiment the frypan is displayed fixedly in the game scene and the egg is displayed moving relative to the frypan. Consequently, there is a need to display a movement of the egg in a direction reverse to the slide direction of the portable game apparatus. Also, the output value (INz) of the Z-axis contact switch 32 and the impact input flag (FS) are not utilized.

In the kitchen-knife table, the output value (INx, INy) of the XY-axis acceleration sensor 31 is defined to be utilized in calculating a change amount of a cabbage X-and-Y coordinate (CAx, CAy). Due to this, when a slide is input to the portable game apparatus, the display position of the cabbage is varied to provide display and control as if the relative position of the cabbage and the kitchen knife were varied. Incidentally, in the kitchen-knife table, the correction ratio is defined minus value similarly to the egg-jumping table. This is because, in the present embodiment, the kitchen knife is fixedly displayed in the game scene. In order to display the cabbage moving relative to the kitchen knife, there is a need to display the cabbage moving in a direction reverse to a

slide direction of the portable game apparatus. Also, the output value (INz) of the Z-axis

contact switch 32 is utilized in determination in the cabbage cutting process, and the

impact input flag (FS) is defined not to be utilized.

Figure 54 is a memory map of the work RAM 26. The work RAM 26 stores

temporary data to be used upon executing the game program by the CPU 21. Specifically,

included are an acceleration-sensor output value memory area 263a, an impact input flag

memory area 263b, an egg data memory area 263c and a cabbage data memory area 263d.

The data stored in the acceleration-sensor output value memory area 263a and

impact input flag memory area 263b is similar to the first embodiment, omitting

explanation.

The egg data memory area 263c stores data of egg X coordinate (Ex), Y coordinate

(Ey), height (Eh) and broiling conditions (Ef). The cabbage data memory area 263d

stores data of cabbage X coordinate (CAx), Y coordinate (CAy) and cut conditions

(CAc).

The memory map of the display RAM is similar to Figure 18 in the first

embodiment, omitting explanation.

Hereunder, a flow of game program process will be explained with reference to

Figure 55 to Figure 59.

Figure 55 is a main routing flowchart. When a cartridge 30 is inserted in the

portable game apparatus main body 10 and the power to the portable game apparatus

main body 10 is turned on, a main routine shown in Figure 55 is started. Although in the

third embodiment, 0G position set process or neutral-position set process may be made as

in the first embodiment, it is omitted for the sake of simplifying explanation.

First, in step S71 a sensor output read process is performed similarly to Figure 31

of the first embodiment to read an output value of the XY-axis acceleration sensor 31 and

Z-axis contact switch 32 through the sensor interface 33 (correction by 0G position data and neutral position data is omitted). After the step S71, it is determined in step S72 whether the button A (operation switch 13b) is pressed or not. If in the step S72 the button A is pressed is determined, the process advances to step S73 to make reference to Figure 57 and perform a kitchen knife space process hereinafter described, then the process proceeds to step S76.

If in the step S72, the button A is not pressed is determined, the process proceeds to step S74 to determine whether the button B (operation switch 13c) is pressed or not. If the B button is not pressed is determined in the step S74, the process advances to step S76. If the button B is pressed is determined in the step S74, the process advances to step S75 to perform a frypan space process hereinafter described with reference to Figure 56, then the process advances to step S76.

It is determined in the step S76 whether the game is over is not. Specifically, game over determination is made under a proper condition as suited to a game content, such as game over when a predetermined time has elapsed. If no game over is determined in the step S76, the process returns to the step S71. If game over is determined in the step S76, the main routine is ended.

Figure 56 is a frypan space process flowchart. First, in step S771 reference is made to the frypan table to make a change process to the egg X coordinate (Ex) and Y coordinate (Ey). After the step S771, in step S772 an egg jump process is made that is hereinafter described with reference to Figure 58. After the step S772, in step S773 processing is made to increase the egg-broil condition (Ef) by 1. After the step S773, it is determined in step S774 whether the egg-broil condition (Ef) becomes 100 or greater or not. If it is determined that the egg-broil condition (Ef) is smaller than 100, the frypan space process is ended. If the egg-broil condition (Ef) is 100 or greater is determined, the

process advances to step S775 to perform an egg success process. In the egg success process, a scene, e.g., of completing egg cooking is displayed and score-adding process is made. After the step S775, the frypan space process is ended.

Figure 57 is a kitchen-knife space process flowchart. First, in step S741 reference is made to the kitchen-knife table to perform a change process to the cabbage X coordinate (CAx) and cabbage Y coordinate (CAy). After the step S741, in step S742 a cabbage cut process is made that is hereinafter described with reference to Figure 59. After the step S742, it is determined in step S743 whether the cabbage cut ratio (CAc) becomes 100 or greater or not. If the cabbage cut ratio (CAc) is smaller than 100 is determined, the kitchen-knife space process is ended. If the cabbage cut ratio (CAc) is 100 or greater is determined, the process proceeds to step S774 to perform a cabbage success process. In the cabbage success process, a scene, e.g. of completing cabbage cutting is displayed and score-adding process is made. After the step S774, the kitchen-knife space process is ended.

Figure 58 is an egg jump process flowchart. First, it is determined in step S772a whether there is an output of the Z-axis contact switch or not (e.g. whether there is an impact input in the Z-axis direction or not). If it is determined in the step S772a that there is no output of the Z-axis contact switch, the egg jump process is ended. If there is an output of the Z-axis contact switch is determined in the step S772a, in step S772b display is made jumping the egg. After the step S772b, in step S772c the egg is set in height (Eh) to CH (predetermined value). After the step S772c, in step S772d a sensor output read process is made similarly to Figure 31 of the first embodiment, thereby reading an output of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 through the sensor interface 33 (correction by 0G position data and neutral position data is omitted). After the step S772d, in step S772e reference is made to the egg jump table to perform a change

process to the egg X coordinate (Ex) and egg Y coordinate (Ey). The step S772e, in step S772f processing is made to decrease the egg height (Eh) by 1. After the step S772f, in step S772g processing is made to display based on the egg X coordinate (Ex), Y coordinate (Ey) and height (Eh). After the step S772g, it is determined in step S772h

5   whether the egg has landed or not, i.e. the egg height (Eh) has become 0 or not. If the egg has not landed is determined in the step S772h, the process returns to the step S772d. If the egg has landed is determined in the step S772h, it is determined in step S772a whether an egg landing position is within the frypan or not. If determined within the frypan, then in step S772j a jump success process has made and then the egg jump process is ended. In

10   the jump success process, for example, music of success is generated while displaying "SUCCESS" and score-adding process is made. Where it is determined in the S772I that the egg landing position is outside the frypan, in step S772k a jump failure process is made and then the egg jump process is ended. In the jump failure process, for example, music of failure is generated while displaying "FAILURE" and processing is made to

15   render the egg-broil condition (Ef) 0 (re-frying egg cooking).

Figure 59 is a cabbage cut process flowchart. First, it is determined in step S742a whether there is an output of the Z-axis contact switch or not (i.e. whether there is a movement input in the Z-axis direction or not). If there is no output of the Z-axis contact switch is determined in step S742a, the cabbage cut process is ended. If there is an output

20   of the Z-axis contact switch is determined in the step S742a, it is determined in step S742b whether there is a cabbage below the kitchen knife or not. If it is determined in the step S742b that there is no cabbage below the kitchen knife, the cabbage cut process is ended. If there is a cabbage below the kitchen knife is determined in the step S742b, in step S742c a display process is made (display of cutting a constant amount of cabbage).

25   After the step S742c in step S742d processing is made to increase the cabbage cut ratio

(CAc) by 1 and then the cabbage cut process is ended.

(Fourth Embodiment)

Next, a fourth embodiment of the invention will be explained with reference to

5   Figure 60 to Figure 66. Figure 60 illustrates a concept view of a game space and example

of a game scene of a plurality of portable game apparatuses. This game shares a game

space through communication between the portable game apparatuses so that a plurality

of players can enjoy a game while competing (or cooperating) a game similar to the first

embodiment. The game space has a maze plate that is common to the portable game

10   apparatuses 10 and 40 so that the game images on the portable game apparatus 10 and

portable game apparatus 40 are on the basis of the same game space data (note that the

range of sight is different between the portable game apparatuses). On the LCD of the

first portable game apparatus 10 a range 12 shown by the one-dot chain line is displayed.

On the LCD of the second portable game apparatus 40, a range 42 shown by the dotted

15   line is displayed. Similarly to the first embodiment, the tilt of the maze plate as a game

space is simulated in accordance with a tilt of the portable game apparatus. However, in

the present embodiment, simulation of a maze plate tilt is made by a value combining a

tilt of the portable game apparatus 10 and a tilt of the portable game apparatus 40

(simulation of a maze plate tilt may be by a tilt of one portable game apparatus). A player

20   on the portable game apparatus 10 would try to operate the tilt of the maze plate by tilting

the portable game apparatus 10 in order to manipulate his or her own ball 61a. On the

other hand, a player on the portable game apparatus 40 would try to operate the tilt of the

maze plate by tilting the portable game apparatus 40 in order to manipulate his or her own

ball 61b. Thus, they are difficult to tilt the maze plate in line with their intentions,

25   providing enjoy for a more complicated game. Incidentally, in this embodiment, a

- 55 -

communication cable 50 is used to communicate between the two portable game apparatuses. However, communication means such as wireless or portable phone may be utilized.

The program ROM of the fourth embodiment stores data almost similar to that of the program ROM (Figure 16) of the first embodiment. However, further stored in a game program memory area a map confirming program hereinafter described with reference to Figure 63 and Figure 64 and a communication interrupt program hereinafter described with reference to Figure 65 and Figure 66, in addition to those of the first embodiment.

Among the programs stored in the game program memory area, the main program, the map confirming program and the communication interrupting program are different between the portable game apparatus 10 and the portable game apparatus 40. This is because to perform communication processing using the portable game apparatus 10 as a master unit and the portable game apparatus 40 as a slave unit, the detail of which will be hereinafter described with reference to Figure 61 to Figure 66.

The work RAM of the fourth embodiment stores data almost similar to that of the work RAM 17 of the first embodiment. However, a composite data memory area is further included in addition to those of the first embodiment. The composite data memory area stores a composite value of an output value of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 of the portable game apparatus 10 and an output value of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 of the portable game apparatus 40.

The memory maps of the display RAM and backup RAM are similar to those of Figure 18 and Figure 19 of the first embodiment, omitting explanation.

Hereunder, a flow of a game program process will be explained with reference to

Figure 61 to Figure 66.

First, Figure 61 is a main routine flowchart to be executed in the portable game apparatus 10. Although in this embodiment the 0G set process, neutral-position set process and impact-input wave generation process are omitted for the sake of simplifying explanation, these processes may be added similarly to the first embodiment.

First, in step S81p a game-map select process is performed similarly to Figure 30 of the first embodiment. After the step S81p, in step S82p reference is made to Figure 63 to perform a master-machine-map confirming process hereinafter described with reference to Figure 63. After the step S82p, the process advances to step S83p.

Steps S83p to S85p are a main loop to be repeatedly processed until game-over or game-clear is reached. In step S83p, required data is written to the display RAM 25 based on the data of the work RAM 26 so that game scenes are displayed on the LCD 12 based on the data stored on the display RAM 25. In step S84p, an each-object moving process (wave moving process is omitted) is made similarly to that of Figure 32 to Figure 36 of the first embodiment, thus processing to move the player character and NPC. After the step S84p, in step S85p a collision process is performed similarly to that of Figure 37 of the first embodiment, thus processing to collide the player character with an NPC or the like. After the step S85p, in step S86p a screen scroll process is made similarly to that of Figure 40 of the first embodiment.

Figure 62 is a main routine flowchart to be executed in the portable game apparatus 40. Although in this embodiment the 0G set process, neutral-position set process and impact-input wave generation process are omitted in order for simplifying explanation, these processes may be added similarly to the first embodiment.

First, in step S81c a game-map select process is made similarly to that of Figure 30 of the first embodiment. After the step S81c, in step S82c a slave-machine map

confirming process is performed that is hereinafter described with reference to Figure 64. After the step S82c, the process advances to step S83c.

Steps S83c to S88c are a main loop to be repeated until game-over or game-clear is reached. First, in step S83c required data is written to the display RAM 25 on the basis of the data of the work RAM 26 so that game scenes are displayed on the LCD 12 on the basis of the data stored on the display RAM 25. After the step S83c, in step S84c a sensor output read process is made similarly to that of Figure 31 of the first embodiment. This reads an output value of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 through the sensor interface 33 (correction by 0G position data and neutral position data is omitted). After the step S84c, in step S85c an interrupt signal and the acceleration-sensor output value data (INx, INy, INz) read out in the former step S84 and stored to the work RAM 26 are transmitted to the portable game apparatus 10. The portable game apparatus 10 receives the interrupt signal and starts a master-machine communication interrupt process hereinafter described with reference to Figure 65. After the step S85c, in step S86c an each-object moving process (wave moving process is omitted) is performed similarly to that of Figure 32 to Figure 36 of the first embodiment, thereby performing a moving process for the player character and NPC. After the step S86c, in step S87c a collision process is performed similarly to that of Figure 37 of the first embodiment, thus processing to collide the player character with an NPC or the like. After the step S87c, in step S88c a screen scroll process is made similarly to that of Figure 40 of the first embodiment.

Figure 63 is a master-machine map confirmation process flowchart to be executed in the portable game apparatus 10. First, in step S87p1 the map number data stored on ones own work RAM 26 is transmitted to the portable game apparatus 40. After the step S87p1, in step S87p2 data transmission and reception is made. Specifically, received is

the map number data transmitted from the portable game apparatus 40 in a step S87c3 of a slave-machine map confirmation process hereinafter described with reference to Figure 64. If data reception is determined in step S87p3, it is then determined in step S87p4 whether the own map number data agrees with the map number data of the portable game apparatus 40 received in the former step S87p2 or not. If agreement of the map number data is determined in step S87p4, the master-machine map confirmation process is ended. If no agreement of the map number data is determined in the step S87p4, the process returns to the game map select process in step S81p of the main routine of Figure 61.

Figure 64 is a slave-machine map confirmation process flowchart to be executed in a portable game apparatus 40. First, in step S87c1 data transmission and reception is made. Specifically, received is the map number data transmitted from the portable game apparatus 10 in step S87p1 of the master-machine map confirmation process of Figure 63. If data reception is determined in step S87c2, in step S87c3 the map number data stored on ones own work RAM 26 is transmitted to the portable game apparatus 10. After the step S87c3, it is determined in step S87c4 whether the own map number data agrees with the map number data of the portable game apparatus received in the former step S87c1 or not. If agreement of the map number data is determined in step S87c4, the slave-machine map confirmation process is ended. If no agreement of the map number data is determined in the step S87c4, the process returns to the game map select process in step S81c of the main routine of Figure 62.

Figure 65 is a master-machine communication interrupt process flowchart to be executed in the portable game apparatus 10. This process is started by an interrupt signal transmitted in the step S85c of the main routine for the portable game apparatus 40 shown in Figure 62. First, in step S91p data transmission and reception is made. Specifically, received is an acceleration-sensor output value of the portable game apparatus 40

transmitted in the step S85c of the main routine for the portable game apparatus 40 shown

in Figure 62. After the step S91p, in step S92p a sensor output read process is made

similarly to that of Figure 31 of the first embodiment, thereby reading an output value of

the XY-axis acceleration sensor 31 and Z-axis contact switch 32 through the sensor

5   interface 33 (correction by 0G position data and neutral position data is omitted). After

step S92p, in step S93p composition is made of an acceleration-sensor output value of the

portable game apparatus 40 received in the former step S91p and an acceleration-sensor

output value of portable game apparatus 10 read out in the former step S92p. Here,

composition may be by a calculation process of mere addition, or by calculation of a

10   composite value from two values through a complicate calculation formula, e.g. adding

two values together with weighting. After the step S93p, in step S94p an interrupt signal

and the composite data calculated in the former step S93p are transmitted to the portable

game apparatus 40.

Figure 66 is a slave-machine communication interrupt flowchart to be executed in

15   the portable game apparatus 40. This process is started according to an interrupt signal

transmitted in step S94p of the master-machine communication interrupt process of

Figure 65. In step S91c the composite data is received from the portable game apparatus

10, and the process is ended.

Although in the above embodiment the portable game apparatus was provided

20   with detecting means, detecting means may be provided on a controller of a home-use

game machine, personal computer, or business-purpose game machine as shown in

Figure 67. In this case, a player can control a game space displayed on a display device,

such as television receiver, by tilting or applying a movement or impact to the controller.

For example, as shown in Figure 68 tilting the controller provides display of tilting a plate

25   as a game space on the display device wherein simulation is provided to roll a ball on the

plate. Simulation is such that tilting the controller to the right provides a tilt of the plate to the right to roll the ball to the right whereas tilting the controller to the left provides a tilt of the plate to the left to roll the ball to the left.

Although in the above embodiments the acceleration sensor was provided on the cartridge, the acceleration sensor may be provided on the side of the portable game apparatus main body. In the case of providing an acceleration sensor on the side of the portable game apparatus main body, there is no need to provide an acceleration sensor for each cartridge, reducing cost. Also, the information storage medium used for the portable game apparatus is not limited to a cartridge but may be an IC card, such as a PC card.

Although in the above first embodiment the neutral position data was stored on the work RAM 26 and set up each time of game play, it may be stored on the backup RAM 35 so that the same data can be utilized in next-round of game play.

Although in the above first embodiment the neutral position was determined by a player, neutral position data may be previously stored in a game program so that it can be utilized. Also, a plurality of neutral position data may be stored so that a player can select any of them.

In the first embodiment, the game characters employed only the player character (ball) and enemy character (tortoise). However, in addition to them, it is possible to appear NPC (non-player character), such as ally characters, assisting the player character or neutral characters. These NPCs, although self-controlled according to a game program (NPC not self-controlled may be provided), may be moved or deformed according to an operation (tilt, movement or impact input) by a player.

Although in the above first embodiment game-space control was based only on an output of the acceleration sensor, there may provided a portion of a game space to be controlled according to an operation switch. For example, it is possible to contemplate

- 61 -

such a game that in a pin ball game a flipper operates when pressing an operation switch while controlling a pin ball board as a game space by tilting or swinging the portable game apparatus.

Also, in a game so-called "fall game" wherein fall objects are piled up so that score is calculated according to a state of piling up, it is possible to contemplate such a game that an object is changed in direction by operation switches or moved at high speed due to impact input or deformed due to movement input in the Z-axis direction while controlling the game space by tilting or swinging the portable game apparatus

Although in the above first embodiment the game characters were moved in accordance with a tilt of the portable game apparatus (i.e. tilt of the maze plate as a game space), they may be moved according to a movement or impact to the portable game apparatus. For example, it is possible to contemplate to provide display and control such that, when the portable game apparatus is slid, simulation is given to move a maze plate wall similarly, moving a game character contacting the wall as if it were pressed by the wall.

Although in the above embodiment the player character (ball) itself was displayed moving, the player character may be displayed fixedly and the game space be scrolled so that the player character is displayed moving relative to the game space.

Although in the above fourth embodiment the two players made the same control to tilt the maze plate, the two players may do individual control. For example, it is possible to contemplate such a game that one player tilts the portable game apparatus to control and tilt a maze plate whereas the other player inputs movement in the Z-axis direction to the portable game apparatus to cause a game character to jump or applies an impact in the XY-axis direction to generate and control waves.

In the above fourth embodiment, the portable game apparatus 10 stored the

master-machine program and the portable game apparatus 40 a slave-machine program, in respect of the main, map confirmation and communication interrupt programs. Instead, both master-machine program and slave-machine program may be stored on each of the portable game apparatus 10 and the portable game apparatus 40 so that setting can be

5   made as to which one is used as a master or slave unit prior to a start of a game and the program be selected according to such setting.

Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited

10   only by the terms of the appended claims.

WHAT IS CLAIMED IS:

1. A game system having in a related fashion, to a game apparatus having game program storage means storing a game program and processing means for executing the game program, display means to display an image based on a result of processing by the processing means, comprising:

a housing to be held by a player; and

change-state detecting means provided related to said housing and for detecting at least one of an amount and a direction of a change applied to said housing, wherein

said game program storage means stores game space data including image data to display a space for game play, a display control program to cause said display means to display a game space based on the game space data; and

a simulation program for simulating based on an output of said change-state detecting means such that a state of the game space is changed related to at least one of a change amount and a change direction applied to said housing.

2. A game system according to claim 1, wherein said change-state detecting means is to detect, as a change amount and direction, at least one of an amount and a direction of a tilt applied to said housing, and

said simulation program simulating related to the at least one of an amount and a direction of a tilt applied to said housing such that the game space is put into a tilt state.

3. A game system according to claim 1, wherein said change-state detecting means detects, as a change amount and direction, at least one of an amount and a direction of a movement applied to said housing, and

said simulation program simulating related to the at least one of an amount and a direction of a movement applied to said housing such that the game space is put into a tilt state.

4. A game system according to claim 1, wherein said change-state detecting means detects, as a change amount and direction, at least one of an amount and a direction of an impact applied to said housing, and

said simulation program simulating related to the at least one of an amount and a direction of an impact applied to said housing such that the game space is put into a tilt state.

5. A game system according to any of claims 1 to 4, wherein said change-state detecting means is for detecting both of an amount and a direction of a change applied to said housing, and

said simulation program simulating related to the both of an amount and a direction of an impact applied to said housing such that the game space is put into a tilt state.

6. A game system according to claim 1, wherein said housing is a housing of said game apparatus, and

said game apparatus being a portable game apparatus having said display means provided integral on one main surface of said housing.

7. A game system according to claim 6, wherein said game program storage means is accommodated in a cartridge and detachably loaded in said housing of said portable game apparatus, and

said change-state detecting means detecting at least one of an amount and a direction of a change applied to said housing of said portable game apparatus when accommodated in said cartridge and said cartridge is loaded in said housing of said portable game apparatus.

8. A game system according to claim 7, wherein said change-state detecting means is for detecting an operation as a tool due to a change state applied to said housing of said

portable game apparatus,

said game program storage means including a character data storage section to display a moving character movable on the game space,

the game space data being image data to provide display associating a tool having

5   a function of controlling a movement of the moving character displayed on the game space, and

said game program storage means including a character control program to read out a moving character stored in said character data storage section and make processing related to at least one of a change amount and a change direction applied to said housing

10   based on an output of said change-state detecting means such that a display state of the moving character is under control of the tool.

9. A game system according to claim 1, wherein said game program storage means includes a character data storage section to display a moving character movable on the game space, and

15   said game program storage means including character control program to read out a moving character stored in said character data storage section and make control related to at least one of a change amount and a change direction applied to said housing based on an output of said change-state detecting means such that a display state of the moving character changes.

20   10. A game system according to claim 1, wherein said game program storage means further includes a non-player character data storage section to display a non-player character to make a first action on the game space according to a predetermined program irrespectively of an operation by the player, and

said simulation program controlling such that the non-player character makes a

25   first action previously determined by a program when any of change states in amount and

direction is not detected by said change-state detecting means, and such that the non-player character makes in addition to the first action a second action related to at least one of an amount and a direction of a change based on an output of said change-state detecting means when at least one of change states in amount and direction is detected by said

5   change-state detecting means.

11. A game system according to claim 1, wherein said game program storage means includes a character data storage section to display a moving character movable on the game space,

the game space data including data to display a particular area defined such that,

10   when the moving character moves on the game space, the moving character is different in action from that in other area,

said simulation program controlling a display state of the moving character related to at least one of an amount and a direction of a change applied to said housing based on an output of said change-state detecting means, and display-controlling, when the moving

15   character moves on the game space, the moving character is different in action from that in other area.

12. A game system according to claim 1, wherein the game space data includes space data to display a greater game space than a display area to be displayed by said display means,

20   the display control program being to display on said display means image data of a part of the game space existing in a range of the display area of the game space, and

said simulation program simulating a state of only the game space existing in the display area based on at least one of an amount and a direction of a change in an output of said change-state detecting means.

25   13. A game system according to claim 1, wherein said change-state detecting

means detects as a change amount a moving amount of said housing and as a change direction a moving direction,

the game space data including space data to display a game space greater than a display area of said display means, and

5   the display control program displaying on said display means a space area of a part of a game space corresponding to the display area, and gradually moving the display area of the game space in the moving direction by an area corresponding to the moving amount according to a movement of said housing.

14. A game system according to claim 1, wherein said game apparatus has 10   operating means to be operated by a player on one main surface of said housing, and

said simulation program changing a state of the game space in a manner of simulation based on a detection output of said change-state detecting means and an operating state of said operating means.

15. A game information storage medium storing a game program and detachably 15   loaded in a game system structured by operating means having display means in a related manner and including a housing to be held by a player, change-state detecting means provided related to the housing and for detecting at least one of an amount and a direction of a change applied to the housing, and processing means to display on the display means an image obtained by processing a program, comprising

20   game space data including image data to display a space for game play;

a display control program to cause said display means a game space based on the game space data; and

a simulation program to provide simulation based on an output of said change-state detecting means such that a state of the game space is changed related to at least one 25   of an amount and a direction of a change applied to said housing.

16. A game information storage medium storing a game program and detachably loaded in a portable game apparatus including a housing integrally having display means to be held by a player, and processing means to display on the display means an image obtained by processing a program, wherein

a change-state detecting means is provided related to one of the portable game apparatus and the game information storage medium and for detecting at least one of an amount and a direction of a change applied to one of a housing of the portable game apparatus and the game information storage medium, the game information storage medium comprising:

game space data including image data to display a space for game play;

a display control program to cause said display means to display a game space based on the game space data; and

a simulation program to provide simulation based on an output of said change-state detecting means such that a state of the game space is changed related to at least one of an amount and a direction of a change applied to said housing.

17. A game information storage medium according to claim 16, wherein said change-state detecting means is for detecting both of an amount and a direction of a change applied to said housing, and

said simulation program simulating such that a state of the game space is changed related to the both of an amount and a direction of a change applied to said housing.

18. A game system structured at least by two game apparatuses, wherein

the two game apparatuses each have game program storage means to store a program, processing means to execute a game program, and a housing to be held by a player, and in a related fashion display means to display an image based on a result of processing by said processing means,

- 69 -

at least one of the two game apparatuses being provided related to said housing

and having change-state detecting means to detect at least one of an amount and a

direction of a change applied to the housing,

the game system further having data transmitting means connected to the two

5   game apparatuses and for transmitting mutually-related data to the game apparatus on the

opposite side,

the respective of the game program storage means of the two game apparatuses

having the following:

game space data including image data to display a space for game play; and

10   display control programs to cause said display means to display a game space

based on the game space data, wherein

said game program storage means of at least the other of said two game

apparatuses further including a simulation program to provide simulation based on an

output of said change-state detecting means of said one game apparatus transmitted

15   through said data transmitting means such that a state of the game space of the other of

said game apparatuses is changed related to at least one of an amount and a direction of a

change applied to said housing of one of said game apparatuses.

19. A game system according to claim 18, wherein said change-state detecting

means are respectively provided on said two game apparatuses, and

20   the respective of said game program storage means of said two game apparatuses

including a simulation program to provide simulation based on an output of said

change-state detecting means of said one game apparatus such that a state of the game

space of said the other game apparatus is changed related to at least one of an amount and

a direction of a change applied to said housing of said one game apparatus.

25   20. A game system according to claim 18, wherein the game space data stored in

- 70 -

said game program storage means of said one game apparatus and the game space data stored in said game program storage means of said the other game apparatus are selected same game space data,

the simulation program of said one game apparatus changing a state of the game space of said one game apparatus correspondingly to a state of the other game space to be simulated by the game space control program, and

the simulation program of said other game apparatus changing a state of the game space of said the other game apparatus correspondingly to a state of one game space to be simulated by the game space control program.

21. A game control method for a game apparatus including a housing to be held by a player and change-state detecting means provided related to the housing and for detecting at least one of an amount or a direction of a change applied to the housing, comprising the steps of:

(a) displaying a game space according to a game program; and

(b) simulating based on an output of said change-state detecting means such that a state of the game space is changed related to at least one of an amount and a direction of a change applied to said housing:

ABSTRACT OF THE DISCLOSURE

A game system includes a housing to be held by a player. The housing incorporates an XY-axis acceleration sensor to detect an acceleration in an X-axis and Y-axis direction and a Z-axis contact switch to detect an acceleration in a Z-axis direction. These sensor and switch detect at least one of an amount (e.g. tilt amount, movement amount, impact amount or the like) and a direction (e.g. tilt direction, movement direction, impact direction or the like) of a change applied to the housing. A simulation program provides simulation such that a state of a game space is changed related to at least one of the amount and direction of the change applied to the housing.

FIG. 1

30

10

12

11

15

14

13a

13c

13b

13f   13e

FIG. 2

Z-AXIS

Y-AXIS

30

10

11

12

X-AXIS

FIG. 3

FIG. 4

FIG. 5

FIG. 6

32

SIDE VIEW

36

322

321

323

X-AXIS
DIRECTION

324a   324

TOP VIEW

36

321

322a   323a

322   323

324

FIG. 7

36

322   323

321

324

FIG. 8

FIG. 9

Y-AXIS

X-AXIS
+ DIRECTION

X-AXIS
- DIRECTION

X-AXIS

Y-AXIS
+ DIRECTION

Y-AXIS
- DIRECTION

FIG. 10

FIG. 11

Y-AXIS

X-AXIS
+ DIRECTION

X-AXIS
- DIRECTION

X-AXIS

Y-AXIS
- DIRECTION

Y-AXIS
+ DIRECTION

FIG. 12

FIG. 13

VIRTUAL MAP

SLIDE

FIG. 14

FIG. 15

IMPACT INPUT

WAVES

61

62b

62a

FIG. 16

PROGRAM ROM

OBJECT CHARACTER DATA

—34a

MAP DATA

—34b

ACCELERATION SENSOR OUTPUT CONVERSION TABLE

RECOMMENDED POSITION SET TABLE
GAME MAP SELECT TABLE

PLAYER CHARACTER MOVING TABLE

IN-AIR TABLE
ON-FLOOR TABLE
ON-ICE TABLE
UNDER WATER TABLE

NPC MOVING TABLE

NORMAL POSITION TABLE
UPSIDE-DOWN POSITION TABLE

—34c

GAME PROGRAM

MAIN PROGRAM
OG SET PROGRAM
NEUTRAL POSITION SET PROGRAM
RECOMMENDED POSITION SET PROGRAM
GAME MAP SELECT PROGRAM
SENSOR OUTPUT READ PROGRAM
OBJECT MOVING PROGRAM
COLLISION PROGRAM
SCREEN SCROLLING PROGRAM
NPC SELF-CONTROL MOVEMENT PROGRAM
OTHER PROGRAMS

—34e

FIG. 17

WORK RAM

NEUTRAL POSITION DATA

NEUTRAL POSITION DATA X (NPx)
NEUTRAL POSITION DATA Y (NPy)
NEUTRAL POSITION DATA Z (NPz)

26a

ACCELERATION SENSOR OUTPUT VALUES

ACCELERATION SENSOR OUTPUT X (INx)
ACCELERATION SENSOR OUTPUT Y (INy)
Z-AXIS CONTACT SWITCH OUTPUT VALUE (INz)

26b

IMPACT INPUT FLAG

IMPACT INPUT FLAG (FS)

26c

CAMERA COORDINATES OF MAP SELECT SCENE

CAMERA X COORDINATE (Cx)
CAMERA Y COORDINATE (Cy)

26e

GAME MAP NUMBER

GAME MAP NUMBER (MN)

26f

CHARACTER 1

X MOVING ACCELERATION (Ax)
Y MOVING ACCELERATION (Ay)
Z MOVING ACCELERATION (Az)

X MOVING ACCELERATION CHANGE AMOUNT (dAx)
Y MOVING ACCELERATION CHANGE AMOUNT (dAy)
Z MOVING ACCELERATION CHANGE AMOUNT (dAz)

X VELOCITY (Vx)
Y VELOCITY (Vy)
Z VELOCITY (Vz)

X COORDINATE (X)
Y COORDINATE (Y)
Z COORDINATE (Z)

LAST-TIME X COORDINATE (Px)
LAST-TIME Y COORDINATE (Py)
LAST-TIME Z COORDINATE (Pz)

CURRENT POSITION STATUS (SP)

POSE NUMBER (PN)

26g1

CHARACTER 2   26g1

CHARACTER 3   26g1

CHARACTER ·   26g·

. . .

FIG. 18

DISPLAY RAM

OBJECT DATA	
OBJECT X COORDINATE	25a
OBJECT Y COORDINATE	
OBJECT CHARACTER ID	
OBJECT POSE NUMBER	

SCROLL COUNTER DATA	
SCROLL COUNTER X COORDINATE (SCx)	25b
SCROLL COUNTER Y COORDINATE (SCy)	

MAP DATA	25c

FIG. 19

BACKUP RAM

OG POSITION DATA	
OG POSITION DATA X (ZGx)	35
OG POSITION DATA Y (ZGy)	

FIG. 20

GAME MAP SELECT PROCESSING TABLE

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAR CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X(INx)	CHANGE AMOUNT OF CAMERA X COORDINATE (Cx)	×2	—	—	—	—
SENSOR OUTPUT VALUE Y(INy)	CHANGE AOUNT OF CAMERA Y COORDINATE (Cy)	×2	—	—	—	—
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	MAP DECISION	—	—	—	—	—
INPACT INPUT FLAG (FS)	—	—	—	—	—	—

FIG. 21

PLAYER CHARACTER MOVING TABLE (IN-AIR)

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAT CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X(INx)	—	—	—	—	—	—
SENSOR OUTPUT VALUE Y(INy)	—	—	—	—	—	—
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	CHANGE AMOUNT OF Z MOVING ACCELERATION (dAz)	×1	—	—	—	—
INPACT INPUT FLAG (FS)	—	—	—	—	—	—

FIG. 22

PLAYER CHARACTER MOVING TABLE (ON-FLOOR)

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAR CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X(INx)	CHANGE AMOUNT OF X MOVING ACCELERATION (dAx)	×2	Inx>20	40	—	—
SENSOR OUTPUT VALUE Y(INy)	CHANGE AOUNT OF Y MOVING ACCELERATION (dAy)	×2	Iny>20	40	—	—
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	CHANGE AMOUNT OF Z MOVING ACCELERATION (dAz)	×1	—	—	—	—
INPACT INPUT FLAG (FS)	CHANGE AMOUNT OF XY MOVING ACCELERATION (dAx, dAy)	×3	—	—	—	—

FIG. 23

PLAYER CHARACTER MOVING TABLE (ON-ICE)

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAT CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X(INx)	CHANGE AMOUNT OF X MOVING ACCELERATION (dAx)	×3	Inx>20	60	—	—
SENSOR OUTPUT VALUE Y(INy)	CHANGE AOUNT OF Y MOVING ACCELERATION (dAy)	×3	Iny>20	60	—	—
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	CHANGE AMOUNT OF Z MOVING ACCELERATION (dAz)	×1	—	—	—	—
INPACT INPUT FLAG (FS)	CHANGE AMOUNT OF Z MOVING ACCELERATION (dAz)	×5	—	—	—	—

FIG. 24

PLAYER CHARACTER MOVING TABLE (UNDER-WATER)

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAR CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X (INx)	CHANGE AMOUNT OF X MOVING ACCELERATION (dAx)	×1／2	Inx>20	60	—	—
SENSOR OUTPUT VALUE Y (INy)	CHANGE AOUNT OF Y MOVING ACCELERATION (dAy)	×1／2	Iny>20	60	—	—
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	CHANGE AMOUNT OF Z MOVING ACCELERATION (dAz)	×1	—	—	—	—
INPACT INPUT FLAG (FS)	—	—	—	—	—	—

FIG. 25

NPC MOVING TABLE (FOR TORTOISE NORMAL POSITION)

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAT CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X (INx)	CHANGE AMOUNT OF X MOVING ACCELERATION (dAx)	×1／2	Inx<10	0	Inx>20	10
SENSOR OUTPUT VALUE Y (INy)	CHANGE AOUNT OF Y MOVING ACCELERATION (dAy)	×1／2	Iny<10	0	Iny>20	10
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	POSITION INVERSION	—	—	—	—	—
INPACT INPUT FLAG (FS)	—	—	—	—	—	—

FIG. 26

NPC MOVING TABLE (FOR TORTOISE UPSIDE-DOWN POSITION)

	UTILIZATION METHOD	CORRECTION RATIO	PATICULAR CORRECTION CONDITION 1	PATICULAT CORRECTION NUMBER 1	PATICULAR CORRECTION CONDITION 2	PATICULAR CORRECTION NUMBER 2
SENSOR OUTOUT VALUE X(INx)	CHANGE AMOUNT OF X MOVING ACCELERATION (dAx)	×2	Inx>20	40	—	
SENSOR OUTPUT VALUE Y(INy)	CHANGE AOUNT OF Y MOVING ACCELERATION (dAy)	×1	Iny>20	40	—	—
Z-AXIS CONTACT SW OUTPUT VALUE (INz)	POSITION INVERSION	—	—	—	—	—
INPACT INPUT FLAG (FS)	—	—	—	—	—	—

FIG. 27

```
                    ( GAME START )
                          │
                          ▼                    S11
              ╱ FIRST START OR ╲      Yes
             ◁   OG SET REQUEST ?  ▷────────────────┐
              ╲               ╱                      ▼
                    │ No                    ┌──────────────────┐        S12
                    │                       ║  OG SET PROCESS   ║
                    │                       └──────────────────┘
                    │◄─────────────────────────────┘
                    │ Yes
                    ▼                          S14
          ┌──────────────────────┐
          ║ NEUTRAL POSITION SET  ║
          ║       PROCESS         ║
          └──────────────────────┘
                    │
                    ▼                          S17
          ┌──────────────────────┐
          ║   GAME MAP SELECT     ║
          ║      PROCESS          ║
          └──────────────────────┘
                    │
                    ▼
            ╱ MAIN LOOP  ╲
           │  (1/60sec)   │
            ╲            ╱
                    │
                    ▼                       S19
             ╱ DISPLAY  ╲
            │  PROCESS   │◄───────────────────────────────┐
             ╲          ╱                                 │
                    │                                     │
                    ▼          S20                        │        S22
          ┌──────────────────────┐          ┌──────────────────────┐
          ║ SENSOR OUTPUT READ    ║          ║ NEUTRAL POSITION SET  ║
          ║      PROCESS          ║          ║       PROCESS         ║
          └──────────────────────┘          └──────────────────────┘
                    │            S21                      ▲
                    ▼                                     │
            ╱ NEUTRAL POSITION ╲         Yes              │
           ◁  SET REQUEST ?     ▷───────────────────────┘
            ╲                  ╱
                    │ No          S23
                    ▼            Yes
          ╱ IMPACT INPUT FLAG ON? ╲──────────┐        S24
           ╲                      ╱          ▼
                    │ No              ╱ UNDER-WATER? ╲
                    │◄───────────────◁              ▷
                    │            No   ╲            ╱
                    │                       │ Yes
                    │                       ▼              S25
                    │              ┌──────────────────┐
                    │              │  PRODUCE WAVES    │
                    │              └──────────────────┘
                    │◄──────────────────────┘
                    ▼          S26
          ┌──────────────────────┐
          ║  OBJECTS MOVING       ║
          ║     PROCESS           ║
          └──────────────────────┘
                    │          S27
                    ▼
          ┌──────────────────────┐
          ║  COLLISION PROCESS    ║
          └──────────────────────┘
                    │          S29
                    ▼
          ┌──────────────────────┐
          ║  SCREEN SCROLL        ║
          ║     PROCESS           ║
          └──────────────────────┘
                    │
                    ▼
          ┌──────────────────────┐
          │   MAIN LOOP END       │
          └──────────────────────┘
                    │
                    ▼
              ( GAME END )
```

FIG. 28

```
        ┌─────────────┐
        │   OG SET    │
        │   PROCESS   │
        └─────────────┘
               │
               ▼
     ╭─────────────────────╮
     │ POSITION HORIZONTAL TOOTH │──── S121
     │   GROUND AND PRESS   │
     │  OPERATION SWITCH    │
     ╰─────────────────────╯
               │
               ▼
        ┌──────────────┐ ──── S122
        │ OPERATION SWITCH │◄─────┐
        │  INPUT PROCESS   │      │
        └──────────────┘      │
               │               │ No
               ▼               │
          ◇ KEY DEPRESSED? ◇───┘
               │       └── S123
               │ Yes
               ▼
  ┌────────┐  Yes    ◇     Z-AXIS        ◇ ──── S124
  │ ALERT  │◄────────◇ CONTACT SWITCH ON? ◇
  │ SOUND  │         ◇                   ◇
  └────────┘              │
   └ S125                 │ No
                          ▼
        ┌──────────────────────┐
        │ STORE ACCELERATION SENSOR │──── S126
        │  OUTPUT VALUES TO OG     │
        │  POSITION DATA AREA OF   │
        │    BACKUP RAM 35         │
        └──────────────────────┘
                   │
                   ▼
              ┌─────────┐
              │ RETURN  │
              └─────────┘
```

FIG. 29

```
        ┌─────────────────────┐
        │  NEUTRAL POSITION   │
        │    SET PROCESS      │
        └──────────┬──────────┘
                   │
        ╭──────────▼──────────╮
        │  POSITION TO ANGLE  │────── S141
        │  EASY TO PLAY AND   │
        │  PRESS OPERATION    │
        │       SWITCH        │
        ╰──────────┬──────────╯
                   │              ┌─── S142
        ┌──────────▼──────────┐  │
        │  OPERATION SWITCH  ◄├──┐
        │    INPUT PROCESS    │  │
        └──────────┬──────────┘  │
                   │          No │
            ╱──────▼──────╲──────┘
           ╱    SWITCH     ╲
           ╲  DEPRESSED?   ╱
            ╲──────┬──────╱
                   │          └─── S143
                Yes│
        ┌──────────▼──────────────────┐
        │ SUBTRACT 0G POSITION DATA FROM│──── S144
        │ ACCELERATION SENSOR OUTPUT    │
        │         VALUE                 │
        └──────────┬──────────────────┘
        ┌──────────▼──────────────────┐
        │ STORE VALUE CALCULATED IN STEP│─── S145
        │ S144 AND Z-AXIS CONTACT SWITCH│
        │ OUTPUT VALUE TO NEUTRAL       │
        │ POSITION DATA AREA 26a        │
        └──────────┬──────────────────┘
                   │
              ┌────▼────┐
              │ RETURN  │
              └─────────┘
```

FIG. 30

```
        ┌─────────────────┐
        │ GAME MAP SELECT │
        │     PROCESS     │
        └────────┬────────┘
                 │
                 ▼
        ┌─────────────────────┐
        │ INITIALIZE CAMERA   │──── S171
        │ COORDINATES (Cx, Cy)│
        └──────────┬──────────┘
                   │
                   ▼
     ┌──────────────────────────┐
     │ DISPLAY MAP FOR SELECTING│──── S172
     │ GAME MAP BASED UPON CAMERA│
     │      COORDINATES         │
     └────────────┬─────────────┘
                  │
                  ▼
        ┌─────────────────┐
        │ SENSOR OUTPUT READ│──── S173
        │     PROCESS     │
        └────────┬────────┘
                 │
                 ▼
  ┌──────────────────────────────┐
  │ REFER TO CONVERSION TABLE (FOR│
  │     GAME MAP SELECT PROCESS)  │
  │ ┌──────────────────────────┐ │
  │ │ TWICE SENSOR OUTPUT VALUE X (INx)│──── S174
  │ │ IS ADDED TO CAMERA COORDINATE (Cx)│
  │ └──────────────────────────┘ │
  │ ┌──────────────────────────┐ │
  │ │ TWICE SENSOR OUTPUT VALUE (INy) IS│
  │ │ ADDED TO CAMERA COORDINATE (Cy)│
  │ └──────────────────────────┘ │
  └───────────────┬──────────────┘
                  │
                  ▼
```

OUTSIDE OF MAP FOR GAME MAP SELECTION? ──── S175

Yes ──── S176 CORRECT TO DISPLAY END REGION OF MAP FOR GAME MAP SELECTION

No

Z-AXIS CONTACT SWITCH ON? ──── S177

No ──────→

Yes

MAP ICON IN DISPLAY? ──── S178

No ──── S179 ALERT SOUND

Yes

STORE GAME MAP NUMBER OF SELECTED COURSE INTO GAME MAP NUMBER (MN) ──── S181

RETURN

FIG. 31

```
      ┌─────────────────┐
      │  SENSOR OUTPUT  │
      │  READ PROCESS   │
      └─────────────────┘
               │
               ▼
      ┌─────────────────────────┐  ╭─ S201
      │  READ LATCH DATA OF     │
      │  INTERFACE CIRCUIT 33   │
      └─────────────────────────┘
               │
               ▼                      ╭─ S202
  ┌───────────────────────────────────┐
  │  READ ACCELERATION SENSOR OUTPUT  │
  │  VALUE (INx, INy) AND Z-AXIS      │
  │  CONTACT SWITCH OUTPUT VALUE      │
  │  (INz) FORM LATCH DATA,           │
  │  AND STORE SAME TO ACCELERATION   │
  │  SENSOR OUTPUT VALUE MEMORY       │
  │  AREA 26b                         │
  └───────────────────────────────────┘
               │
               ▼          S203
           ◇─────────────◇           No
           │  IMPACT INPUT?  ├────────────┐
           ◇─────────────◇             │
               │                         │
            Yes│   S204                  │  S205
               ▼                         ▼
      ┌─────────────────┐      ┌─────────────────┐
      │ ON IMPACT INPUT │      │ OFF IMPACT INPUT│
      │   FLAG (Fs)     │      │   FLAG (Fs)     │
      └─────────────────┘      └─────────────────┘
               │                         │
               └───────────◄─────────────┘
               │
               ▼                      S206
  ┌───────────────────────────────┐
  │  SUBTRACT OG POSITION DATA    │
  │     IN BACKUP RAM FROM        │
  │  ACCELERATION SENSOR OUTPUT   │
  │  VALUE MEMORY AREA 26 b       │
  └───────────────────────────────┘
               │
               ▼                      S207
  ┌────────────────────────────────────┐
  │ FURTHER CORRECT WITH NEUTRAL       │
  │ POSITION DATA, AND STORE AS INx,   │
  │ INy AND INz INTO ACCELERATION      │
  │ SENSOR OUTPUT VALUE MEMORY AREA    │
  │ 26b                                │
  └────────────────────────────────────┘
               │
               ▼
          ┌─────────┐
          │ RETURN  │
          └─────────┘
```

FIG. 32

```
    OBJECT MOVING
      PROCESS
         │
         ▼                           ┌─ S261
  ┌──────────────────┐
  │ PLAYER CHARACTER │
  │  MOVING PROCESS  │
  └──────────────────┘
         │
         ▼
  ┌──────────────────┐
  │   NPC PROCESS    │
  │      LOOP        │
  └──────────────────┘
         │
         ▼                           ┌─ S262
  ┌──────────────────┐
  │   NPC MOVING     │
  │    PROCESS       │
  └──────────────────┘
         │
         ▼
  ┌──────────────────┐
  │   NPC PROCESS    │
  │    LOOP END      │
  └──────────────────┘
         │
         ▼
     ( RETURN )
```

FIG. 33

```
   PLAYER CHARACTER
    MOVING PROCESS
         │
         ▼                           ┌─ S31
  ┌──────────────────────┐
  │ COPY CURRENT COORDINATES │
  │ AS LAST-TIME COORDINATES │
  └──────────────────────┘
         │
         ▼                           ┌─ S32
  ┌──────────────────────┐
  │ RENDER CHANGE AMOUNT OF │
  │  MOVING ACCELERATION 0 │
  └──────────────────────┘
         │
         ▼                           ┌─ S33
  ┌──────────────────┐
  │  TILT MOVEMENT   │
  │    PROCESS       │
  └──────────────────┘
         │
         ▼                           ┌─ S34
  ┌──────────────────┐
  │  IMPACT MOVEMENT │
  │    PROCESS       │
  └──────────────────┘
         │
         ▼                           ┌─ S35
  ┌──────────────────┐
  │   JUMP PROCESS   │
  └──────────────────┘
         │
         ▼                           ┌─ S36
       ╱         ╲         Yes
      ╱   WAVES   ╲─────────────┐          ┌─ S37
      ╲ PRODUCED? ╱             ▼
       ╲         ╱      ┌──────────────────┐
         │               │  WAVE MOVEMENT   │
         │ No            │    PROCESS       │
         │               └──────────────────┘
         ▼                        │
  ┌──────────────────────┐◄───────┘
  │   CALCULATE MOVING   │       ┌─ S38
  │ ACCELERATION AND VELOCITY │
  └──────────────────────┘
         │
         ▼                           ┌─ S39
  ┌──────────────────┐
  │    CALCULATE     │
  │   COORDINATES    │
  └──────────────────┘
         │
         ▼
     ( RETURN )
```

FIG. 34

```
        ┌─────────────┐
        │ NPC MOVING  │
        │   PROCESS   │
        └──────┬──────┘
               │                              S41
        ┌──────▼────────────────────┐
        │ COPY CURRENT COORDINATES AS│
        │   LAST-TIME COORDINATES    │
        └──────┬────────────────────┘
               │                              S42
        ┌──────▼────────────────────┐
        │   RENDER CHANGE AMOUNT OF   │
        │   MOVING ACCELERATION 0     │
        └──────┬────────────────────┘
               │                              S43
        ┌──────▼────────┐
        │ SELF-CONTROLLED│
        │   MOVEMENT     │
        └──────┬────────┘
               │                              S44
        ┌──────▼────────┐
        ║ TILT MOVEMENT  ║
        ║   PROCESS      ║
        └──────┬────────┘
               │                              S45
        ┌──────▼────────┐
        ║ IMPACT MOVING  ║
        ║   PROCESS      ║
        └──────┬────────┘
               │                              S46
            ◇──▼──◇         Yes
          ◇  WAVES   ◇──────────────┐         S47
          ◇PRODUCED? ◇              │  ┌───────▼───────┐
            ◇─────◇                 │  ║ WAVE MOVEMENT  ║
               │ No                 │  ║   PROCESS      ║
               │                    └──┘───────┬───────┘
               │◄────────────────────────────┘
               │                              S48
        ┌──────▼────────────┐
        │  CALCULATE MOVING   │
        │ ACCELERATION AND VELOCITY│
        └──────┬────────────┘
               │                              S49
        ┌──────▼────────┐
        │   CALCULATE    │
        │  COORDINATES   │
        └──────┬────────┘
               │                              S51
            ◇──▼──◇          Yes
       ◇  Z-AXIS CONTACT ◇─────────────┐      S52
       ◇SWITCH OUTPUT VALUE 1?◇        │ ┌─────▼─────┐
            ◇──────◇                   │ │  POSITION │
               │ No                    │ │ INVERSION │
               │                       └─┘─────┬─────┘
               │◄─────────────────────────────┘
               │
          ┌────▼────┐
          │ RETURN  │
          └─────────┘
```

FIG. 35

```
        ┌─────────────────┐
        │  JUMP PROCESS   │
        └─────────────────┘
                 │
                 ▼                    S351
            ╱─────────╲
           ╱   Z-AXIS  ╲        No
          ╱CONTACT SWITCH╲──────────────────────┐
          ╲OUTPUT VALUE 1?╱                      │
           ╲             ╱                       ▼              S354
            ╲───────────╱                   ╱─────────╲
   S352          │ Yes                     ╱CURRENT POSITION╲   Yes
                 ▼                         ╲STATUS "IN-AIR"? ╱────────┐
      ┌─────────────────────┐              ╲               ╱         │
      │ RENDER CURRENT POSITION│            ╲─────────────╱          ▼         S355
      │   STATUS "IN-AIR"   │                   │ No           ┌──────────────┐
      └─────────────────────┘                   │              │RENDER CHANGE │
                 │              S353             │              │AMOUNT OF Z-AXIS│
                 ▼                               │              │   MOVING     │
   ┌──────────────────────────┐                 │              │ ACCELERATION -1│
   │ RENDER CHANGE AMOUNT OF Z-│                │              └──────────────┘
   │AXIS MOVING ACCELERATION -1│◄───────────────┘                     │
   └──────────────────────────┘                                       │
                 │                                                     ▼
                 ▼                                                 ( RETURN )
             ( RETURN )
```

FIG. 36

```
        ┌──────────────────┐
        │  WAVE MOVEMENT   │
        │     PROCESS      │
        └──────────────────┘
                 │
                 ▼                S361
        ┌──────────────────┐
        │  READ CURRENT    │
        │ POSITION STATUS  │
        └──────────────────┘
                 │            S362
                 ▼
            ╱─────────╲
           ╱ AFFECTED   ╲    No
          ╱ POSITION     ╲─────────────┐
          ╲ BY WAVES?    ╱             │
           ╲           ╱               │
            ╲─────────╱                │
  S363           │ Yes                 │
                 ▼                     │
   ┌──────────────────────────┐        │
   │ ADD CHANGE AMOUNTS OF Z-AXIS│     │
   │ AND Y-AXIS MOVING ACCELERATION│   │
   │ DUE TO AFFECTION OF WAVES TO│     │
   │      EACH OTHER           │       │
   └──────────────────────────┘        │
                 │                      │
                 ▼◄─────────────────────┘
             ( RETURN )
```

FIG. 37

COLLISION
PROCESS

NPC COLLISION
DETERMINATION
LOOP

S271
COLLISION WITH WALL? —Yes→

No

S272
COLLISION
WITH OTHER NPC? —Yes→

No

S275
DETECT CURRENT
POSITION STATUS

S273
GENERATE COLLISION
SOUND

S274
RETURN NPC COORDINATES
TO LAST-TIME
COORDINATES

NPC COLLISION
DETERMINATION
LOOP END

S276
PLAYER CHARACTER
COLLIDED WITH WALL? —Yes→

No

S277
COLLISION SOUND

S278
RETURN PLAYER CHARACTER COORDINATES
TO LAST-TIME COORDINATES

S279
DETECT CURRENT
POSITION STATUS

S281
PLAYER CHARACTER
COLLIDED WITH NPC? —Yes→

No

S282
VANISH NPC

S283
NPCs ALL VANISHED? —No→

Yes

S284
GAME CLEAR PROCESS

S285
PLAYER CHARACTER
FALLEN IN HOLE? —Yes→

No

S286
GAME OVER PROCESS

RETURN

FIG. 38

FIG. 39

FIG. 40

```
        ( SCREEN SCROLL )
        (    PROCESS    )
                │
                ▼
        ╱─────────────────╲  S291
      ╱  PLAYER CHARACTER IS OUT OF SCROLL  ╲  Yes        ╱────────────────╲  S292
     ╱   AREA IN X-AXIS MINUS DIRECTION?     ╲──────────▶╱   LEFT END OF    ╲
     ╲                                       ╱          ╲ ENTIRE GAME MAP?   ╱
      ╲─────────────────╱                     Yes        ╲────────────────╱
                │ No                                            │ No    S293
                │                                    ┌──────────────────────────┐
                │                                    │ DECREASE SCROLL COUNTER X │
                │◀───────────────────────────────── │ COORDINATE (SCx) BY GIVEN │
                │                                    │          AMOUNT           │
                │◀───────────────────────────────────────┘
                ▼
        ╱─────────────────╲  S294
      ╱  PLAYER CHARACTER IS OUT OF SCROLL  ╲  Yes        ╱────────────────╲  S295
     ╱   AREA IN X-AXIS PLUS DIRECTION?      ╲──────────▶╱  RIGHT END OF    ╲
     ╲                                       ╱          ╲ ENTIRE GAME MAP?   ╱
      ╲─────────────────╱                     Yes        ╲────────────────╱
                │ No                                            │ No    S296
                │                                    ┌──────────────────────────┐
                │                                    │ INCREASE SCROLL COUNTER X │
                │◀───────────────────────────────── │ COORDINATE (SCx) BY GIVEN │
                │                                    │          AMOUNT           │
                │◀───────────────────────────────────────┘
                ▼
        ╱─────────────────╲  S297
      ╱  PLAYER CHARACTER IS OUT OF SCROLL  ╲  Yes        ╱────────────────╲  S298
     ╱   AREA IN Y-AXIS MINUS DIRECTION?     ╲──────────▶╱  UPPER END OF    ╲
     ╲                                       ╱          ╲ ENTIRE GAME MAP?   ╱
      ╲─────────────────╱                     Yes        ╲────────────────╱
                │ No                                            │ No    S299
                │                                    ┌──────────────────────────┐
                │                                    │ DECREASE SCROLL COUNTER Y │
                │◀───────────────────────────────── │ COORDINATE (SCy) BY GIVEN │
                │                                    │          AMOUNT           │
                │◀───────────────────────────────────────┘
                ▼
        ╱─────────────────╲  S301
      ╱  PLAYER CHARACTER IS OUT OF SCROLL  ╲  Yes        ╱────────────────╲  S302
     ╱   AREA IN Y-AXIS PLUS DIRECTION?      ╲──────────▶╱  LOWER END OF    ╲
     ╲                                       ╱          ╲ ENTIRE GAME MAP?   ╱
      ╲─────────────────╱                     Yes        ╲────────────────╱
                │ No                                            │ No    S303
                │                                    ┌──────────────────────────┐
                │                                    │ INCREASE SCROLL COUNTER Y │
                │◀───────────────────────────────── │ COORDINATE (SCy) BY GIVEN │
                │                                    │          AMOUNT           │
                │◀───────────────────────────────────────┘
                ▼
            ( RETURN )
```

FIG. 41

(a)

(b)

(c)

FIG. 42

FIG. 43

FIG. 44

(a)

(b)

FIG. 45

PROGRAM ROM

| OBJECT CHARACTER DATA | 342a |

| MAP DATA | 342b |

DATA OF UPHEAVAL POINT IN LAND

| COORDINATE OF UPHEAVAL POINT 1 (PX1,Py1) |
| COORDINATE OF UPHEAVAL POINT 2 (PX2,Py2) |
| COORDINATE OF UPHEAVAL POINT 3 (Px3,Py3) |
| . . . |
| COORDINATE OF UPHEAVAL POINT 12 (Px12,Py12) |

342c

LIMIT VALUES OF SCROLL
(SCxmax, SCymax)

342d

ACCELERATION SENSOR OUTPUT VALUE CONVERSION TABLE
(FOR MOVING VIEW IN LAND UPHEAVAL GAME)

342e

GAME PROGRAM

| MAIN PROGRAM |
| SENSOR OUTPUT READ PROGRAM |
| VIEW MOVEMENT PROGRAM |
| LAND UPHEAVAL PROGRAM |
| TEMPERATURE INCREASE PROGRAM |
| TORTOISE CONTROL PROGRAM |
| OTHER PROGRAMS |

342f

F I G. 46

WORK RAM

ACCELERATION SENSOR OUTPUT VALUES	
SENSOR OUTPUT VALUE X (INx)	
SENSOR OUTPUT VALUE Y (INy)	
Z-AXIS CONTACT SWITCH OUTPUT VALUE (INz)	

~262a

IMPACT INPUT FLAG
IMPACT INPUT FLAG (Fs)

~262b

LAND UPHEAVAL DATA
HEIGHT OF UPHEAVAL POINT 1 (H1)
HEIGHT OF UPHEAVAL POINT 2 (H2)
HEIGHT OF UPHEAVAL POINT 3 (H3)
· · ·
HEIGHT OF UPHEAVAL POINT 12 (H12)

~262c

TEMPERATURE
TEMPERATURE (T)

~262d

TORTOISE CHARACTER DATA

TORTOISE CHARACTER 1

X COORDINATE (X1)
Y COORDINATE (Y1)

LAST-TIME X COORDINATE (Px1)
LAST-TIME Y COORDINATE (Py1)

TORTOISE CHARACTER 2

TORTOISE CHARACTER 3

· · ·

TORTOISE CHARACTER m

~262e

FIG. 47

```
          ┌──────────────┐
          │    START     │
          └──────┬───────┘
                 │
    ┌────────────┤
    │            ▼
    │    ┌─────────────────────┐
    │    │ SENSOR OUTPUT READ  │──── S61
    │    │      PROCESS        │
    │    └──────────┬──────────┘
    │               ▼
    │    ┌─────────────────────┐
    │    │ VIEW MOVEMENT       │──── S62
    │    │      PROCESS        │
    │    └──────────┬──────────┘
    │               ▼
    │    ┌─────────────────────┐
    │    │ LAND UPHEAVAL       │──── S63
    │    │      PROCESS        │
    │    └──────────┬──────────┘
    │               ▼
    │    ┌─────────────────────┐
    │    │ TEMPERATURE INCREASE│──── S64
    │    │      PROCESS        │
    │    └──────────┬──────────┘
    │               ▼
    │    ┌─────────────────────┐
    │    │ CHARACTER CONTROL   │──── S65
    │    │      PROCESS        │
    │    └──────────┬──────────┘
    │               ▼
    │    ┌─────────────────────┐
    │    │  DISPLAY PROCESS    │──── S66
    │    └──────────┬──────────┘
    │               │              S67
    │     No     ◇──┴──◇
    └───────────  GAME OVER?
                ◇───┬───◇
                    │ Yes
                    ▼
          ┌──────────────┐
          │     END      │
          └──────────────┘
```

FIG. 48

VIEW MOVEMENT
PROCESS

REFER TO CONVERSION TABLE (FOR LAND
UPHEAVAL GAME)

CHANGE SCROLL COUNTER X
COORDINATE (SCx) BASE UPON
SENSOR OUTPUT VALUE X(INx)

CHANGE SCROLL COUNTER Y
COORDINATE (SCy) BASE UPON
SENSOR OUTPUT VALUE Y(INy)

S621

S622

$SCx > SCxmax$?

Yes

S623

$SCx = SCxmax$

No

S624

$SCx < 0$?

Yes

S625

$SCx = 0$

No

S626

$SCy > SCymax$?

Yes

S627

$SCy = SCymax$

No

S628

$SCy < 0$?

Yes

S629

$SCy = 0$

No

RETURN

FIG. 49

```
        ┌─────────────────┐
        │  LAND UPHEAVAL  │
        │     PROCESS     │
        └────────┬────────┘
                 │
                 ▼                              S631
          ◇─────────────────◇    No
         ◇  OUTPUT OF Z-AXIS  ◇─────────────────────────────────┐
          ◇ CONTACT SWITCH? ◇                                    │
           ◇───────┬───────◇                                     │
              Yes  │                                             │
                   ▼                    S632                     │
             ◇──────────────◇    No                             │
            ◇ A BUTTON DEPRESSED? ◇──────────┐                  │
             ◇──────┬───────◇                │                  │
               Yes  │                        ▼                  │
                    │              ◇──────────────────◇  S634   │
                    │             ◇ B BUTTON DEPRESSED? ◇  No   │
                    │       S633   ◇────────┬─────────◇────┐    │
                    │                       │ Yes           │    │
                    ▼                       ▼  S635         ▼ S636
        ┌─────────────────┐    ┌─────────────────┐  ┌─────────────────┐
        │ INCREASE HEIGHT OF│  │ INCREASE HEIGHT OF│  │ INCREASE HEIGHT OF│
        │ UPHEAVAL POINTS IN│  │UPHEAVAL POINTS OUT│  │  ALL UPHEAVAL   │
        │ DISPLAY AREA BY 1 │  │OF DISPLAY AREA BY 1│ │  POINTS BY 1    │
        └─────────┬─────────┘  └─────────┬─────────┘  └────────┬────────┘
                  │◄─────────────────────┘                     │
                  │◄────────────────────────────────────────────┘
                  │
                  ▼
             ┌─────────┐
             │ RETURN  │
             └─────────┘
```

F I G. 50

FIG. 51

FIG. 52

FIG. 53

(a)

99

98

(b)

99   98

(c)

99   98   99'

FIG. 54

WORK RAM

ACCELERATION SENSOR OUTPUT VALUES — 263a
- SENSOR OUTPUT VALUE X (INx)
- SENSOR OUTPUT VALUE Y (INy)
- Z-AXIS CONTACT SWITCH OUTPUT VALUE (INz)

IMPACT INPUT FLAG — 263b
- IMPACT INPUT FLAG (Fs)

EGG DATA — 263c
- X COORDINATE (Ex)
- Y COORDINATE (Ey)
- HEIGHT (Eh)
- FIRE (Ef)

CABBAGE DATA — 263d
- X COORDINATE (CAx)
- Y COORDINATE (CAy)
- CUT (CAc)

FIG. 55

START

S71 — SENSOR OUTPUT READ PROCESS

S72 — A BUTTON DEPRESSED?
- Yes → S73 KITCHEN KNIFE SPACE PROCESS
- No ↓

S74 — B BUTTON DEPRESSED?
- Yes → S75 FRYPAN SPACE PROCESS
- No ↓

S76 — GAME OVER?
- No → (loop back to S71)
- Yes → END

FIG. 56

```
        ┌─────────────────┐
        │  FRYPAN SPACE   │
        │    PROCESS      │
        └────────┬────────┘
                 │
   ┌─────────────▼──────────────────────────┐  S771
   │ REFER TO CONVERSION TABLE (FOR FRYPAN)  │
   │ ┌────────────────────────────────────┐ │
   │ │ CHANGE EGG X COORDINATE (Ex) BASE  │ │
   │ │ UPON SENSOR OUTPUT VALUE (INx)     │ │
   │ ├────────────────────────────────────┤ │
   │ │ CHANGE EGG Y COORDINATE (Ey) BASE  │ │
   │ │ UPON SENSOR OUTPUT VALUE (INy)     │ │
   │ └────────────────────────────────────┘ │
   └─────────────┬──────────────────────────┘
                 │
        ┌────────▼────────┐  S772
        │ EGG JUMP PROCESS │
        └────────┬────────┘
                 │
        ┌────────▼────────┐  S773
        │INCREASE FIRE (Ef)│
        │      BY 1        │
        └────────┬────────┘
                 │
                 ▼            S774        S775
            ╱◇◇◇◇◇╲  Yes  ┌──────────────┐
           ╱ (Ef)>= ╲─────▶│ EGG SUCCESS  │
           ╲  100?  ╱      │   PROCESS    │
            ╲◇◇◇◇◇╱        └──────┬───────┘
                 │ No             │
                 ◀────────────────┘
                 ▼
            ┌─────────┐
            │ RETURN  │
            └─────────┘
```

FIG. 57

```
        ┌──────────────────┐
        │  KITCHEN KNIFE   │
        │  SPACE PROCESS   │
        └────────┬─────────┘
                 │
   ┌─────────────▼────────────────────────────────┐  S741
   │ REFER TO CONVERSION TABLE (FOR KITCHEN KNIFE) │
   │ ┌──────────────────────────────────────────┐ │
   │ │ CHANGE CABBAGE X COORDINATE (CAx) BASE    │ │
   │ │ UPON SENSOR OUTPUT VALUE (INx)            │ │
   │ ├──────────────────────────────────────────┤ │
   │ │ CHANGE CABBAGE Y COORDINATE (CAy) BASE    │ │
   │ │ UPON SENSOR OUTPUT VALUE (INy)            │ │
   │ └──────────────────────────────────────────┘ │
   └─────────────┬────────────────────────────────┘
                 │
        ┌────────▼────────────┐  S742
        │ CABBAGE CUT PROCESS │
        └────────┬────────────┘
                 │                S743       S744
                 ▼            ╱◇◇◇◇◇╲  Yes  ┌───────────────┐
                             ╱ (CAc)>= ╲────▶│CABBAGE SUCCESS│
                             ╲  100?   ╱     │   PROCESS     │
                              ╲◇◇◇◇◇╱        └───────┬───────┘
                                 │ No               │
                                 ◀─────────────────┘
                                 ▼
                            ┌─────────┐
                            │ RETURN  │
                            └─────────┘
```

FIG. 58

```
        ┌──────────────────┐
        │    EGG JUMP      │
        │    PROCESS       │
        └────────┬─────────┘
                 │
                 ▼                        ╔═ S772a
            ◇─────────────◇      No
           ╱ Z-AXIS CONTACT ╲─────────────────────┐
           ╲ SWITCH OUTPUT? ╱                      │
            ◇─────────────◇                        │
                 │ Yes                             │
                 ▼                                 │
        ┌──────────────────┐      S772b            │
        │  DISPLAY EGG JUMP │                      │
        └────────┬─────────┘                       │
                 │                                 │
                 ▼                                 │
        ┌──────────────────┐      S772c            │
        │ RENDER EGG HEIGHT │                      │
        │    (Eh) = CH      │                      │
        └────────┬─────────┘                       │
                 │                                 │
                 ▼◄──────────────────────────────┐ │
        ┌──────────────────┐   S772d            │ │
        ║ SENSOR OUTPUT READ ║                   │ │
        ║    PROCESS         ║                   │ │
        └────────┬─────────┘                     │ │
                 │                               │ │
                 ▼                               │ │
  ┌──────────────────────────────────┐          │ │
  │ REFER TO CONVERSION TABLE (FOR    │          │ │
  │        FRYPAN JUMP)               │          │ │
  │  ┌────────────────────────────┐   │          │ │
  │  │ CHANGE EGG X COORDINATE (Ex)│  │          │ │
  │  │ BASED UPON SENSOR OUTPUT    │  │  S772e    │ │
  │  │      VALUE X (INx)          │  │          │ │
  │  └────────────────────────────┘   │          │ │
  │  ┌────────────────────────────┐   │          │ │
  │  │ CHANGE EGG Y COORDINATE (Ey)│  │          │ │
  │  │ BASED UPON SENSOR OUTPUT    │  │          │ │
  │  │      VALUE Y (INy)          │  │          │ │
  │  └────────────────────────────┘   │          │ │
  └────────────────┬─────────────────┘           │ │
                   │                              │ │
                   ▼                              │ │
        ┌──────────────────┐    S772f             │ │
        │ DECREASE EGG HEIGHT │                   │ │
        │    (Eh) BY 1       │                    │ │
        └────────┬─────────┘                      │ │
                 │                                │ │
                 ▼                                │ │
        ┌──────────────────┐    S772g             │ │
        │  DISPLAY PROCESS  │                      │ │
        └────────┬─────────┘                       │ │
                 │            S772h                 │ │
                 ▼                                  │ │
            ◇─────────────◇      No                 │ │
           ╱ EGG LANDING   ╲──────────────────────┘ │
           ╲  (Eh=0)?      ╱                         │
            ◇─────────────◇                          │
                 │ Yes                               │
                 ▼             S772i                 │
            ◇─────────────◇    No                    │
           ╱ EGG LANDING POINT╲─────────┐            │
           ╲ WITHIN FLYING PAN?╱         │            │
            ◇─────────────◇              │            │
                 │ Yes  S772j            │ S772k      │
                 ▼                       ▼            │
        ┌──────────────┐       ┌──────────────┐      │
        │ JUMP SUCCESS │       │ JUMP FAILURE │      │
        │   PROCESS    │       │   PROCESS    │      │
        └──────┬───────┘       └──────┬───────┘      │
               │                      │              │
               ▼◄─────────────────────┘              │
               │◄───────────────────────────────────┘
               ▼
        ┌──────────────┐
        │   RETURN     │
        └──────────────┘
```

FIG. 59

```
        ┌─────────────────┐
        │  CABBAGE CUT    │
        │  PROCESS        │
        └─────────────────┘
                 │
                 ▼                    S742a
            ╱─────────╲
           ╱ Z-AXIS    ╲      No
          ╱ CONTACT     ╲─────────────────────┐
          ╲ SWITCH OUTPUT?                     │
           ╲           ╱                       │
            ╲─────────╱                        │
                 │ Yes                         │
                 ▼                    S742b    │
            ╱─────────╲                        │
           ╱ CABBAGE   ╲       No              │
          ╱ BELOW       ╲──────────────────────┤
          ╲ KITCHEN KNIFE?                     │
           ╲           ╱                       │
            ╲─────────╱                        │
                 │ Yes                         │
                 ▼                             │
        ┌─────────────────┐        S742c       │
        │ DISPLAY PROCESS │                    │
        └─────────────────┘                    │
                 │                             │
                 ▼                             │
        ┌─────────────────┐        S742d       │
        │ INCREASE CABBAGE│                    │
        │ CUT (CAc) BY 1  │                    │
        └─────────────────┘                    │
                 │                             │
                 ▼◄────────────────────────────┘
              ╱──────╲
             │ RETURN │
              ╲──────╱
```

FIG. 60

FIG.61

MASTER MACHINE

```
        ( GAME START )
               │
               ▼
    ┌───────────────────────┐      S81p
    │   GAME MAP SELECT      │
    │      PROCESS           │
    └───────────────────────┘
               │
               ▼
    ┌───────────────────────┐      S82p
    │   MAP CONFIRMATION     │
    │      PROCESS           │
    └───────────────────────┘
               │
               ▼
    ╱   MAIN LOOP    ╲
   │   (1/60sec)     │
    ╲               ╱
               │
               ▼
    ╭───────────────╮            S83p
   │    DISPLAY      │
   │    PROCESS      │
    ╰───────────────╯
               │
               ▼
    ┌───────────────────────┐      S84p
    │   OBJECT MOVEMENT      │
    │      PROCESS           │
    └───────────────────────┘
               │
               ▼
    ┌───────────────────────┐      S85p
    │   COLLISION PROCESS    │
    └───────────────────────┘
               │
               ▼
    ┌───────────────────────┐      S86p
    │   SCREEN SCROLL        │
    │      PROCESS           │
    └───────────────────────┘
               │
               ▼
    ┌───────────────────────┐
    │   MAIN LOOP END        │
    └───────────────────────┘
               │
               ▼
        (  GAME END  )
```

FIG. 62

SLAVE MACHINE

GAME START

GAME MAP SELECT PROCESS — S81c

MAP CONFIRMATION PROCESS — S82c

MAIN LOOP (1/60sec)

DISPLAY PROCESS — S83c

SENSOR OUTPUT READ PROCESS — S84c

TANSMIT INTERRUPT SIGNAL AND SENSOR OUTPUT VALUES TO MASTER MACHINE — S85c

OBJECTS MOVEMENT PROCESS — S86c

COLLISION PROCESS — S87c

SCREEN SCROLL PROCESS — S88c

MAIN LOOP END

GAME END

FIG. 63

MAP CONFIRMATION PROCESS
IN MASTER MACHINE

TRANSMIT MAP NUMBER DATA — S87c1

DATA RECEPTION — S87c2

DATA RECEIVED? — S87c3
No

Yes

MAPS MATCHED? — S87c4
No → RETURN MAP SELECT PROCESS

Yes

RETURN

FIG. 64

MAP CONFIRMATION PROCESS
IN SLAVE MACHINE

DATA RECEPTION — S87c1

DATA RECEIVED? — S87c2
No

TRANSMIT MAP NUMBER DATA — S87c3

Yes

MAPS MATCHED? — S87c4
No → RETURN MAP SELECT PROCESS

Yes

RETURN

FIG. 65

COMMUNICATION INTERRUPT
PROCESS IN MASTER MACHINE

↓

DATA RECEPTION — S91p

↓

SENSOR OUTPUT READ
PROCESS — S92p

↓

COMPOSE ACCELERATION
SENSOR OUTPUT VALUES
AND RECEPTION DATA — S93p

↓

TRANSMIT INTERRUPT SIGNAL
AND COMPOSED DATA TO SLAVE
MACHINE — S94p

↓

RETURN

FIG. 66

COMMUNICATION INTERRUPT
PROCESS IN SLAVE MACHINE

↓

RECEIVE COMPOSED DATA FROM
MASTER MACHINE — S91c

↓

RETURN

FIG. 67

31
32
33

FIG. 68